COMPUTER-BASED MEDICAL GUIDELINES AND PROTOCOLS: A PRIMER AND CURRENT TRENDS

Studies in Health Technology and Informatics

This book series was started in 1990 to promote research conducted under the auspices of the EC programmes' Advanced Informatics in Medicine (AIM) and Biomedical and Health Research (BHR) bioengineering branch. A driving aspect of international health informatics is that telecommunication technology, rehabilitative technology, intelligent home technology and many other components are moving together and form one integrated world of information and communication media. The complete series has been accepted in Medline. Volumes from 2005 onwards are available online.

Volume 139

Recently published in this series

Vol. 138. T. Solomonides et al. (Eds.), Global Healthgrid: e-Science Meets Biomedical Informatics – Proceedings of HealthGrid 2008

Vol. 137. L. Bos, B. Blobel, A. Marsh and D. Carroll (Eds.), Medical and Care Compunetics 5

Vol. 136. S.K. Andersen, G.O. Klein, S. Schulz, J. Aarts and M.C. Mazzoleni (Eds.), eHealth Beyond the Horizon – Get IT There – Proceedings of MIE2008 – The XXIst International Congress of the European Federation for Medical Informatics

Vol. 135. T.B. Grivas (Ed.), The Conservative Scoliosis Treatment – 1st SOSORT Instructional Course Lectures Book

Vol. 134. B. Blobel, P. Pharow and M. Nerlich (Eds.), eHealth: Combining Health Telematics, Telemedicine, Biomedical Engineering and Bioinformatics to the Edge – Global Experts Summit Textbook

Vol. 133. J. Hammer, M. Nerlich and S. Dendorfer (Eds.), Medicine Meets Engineering – Proceedings of the 2nd Conference on Applied Biomechanics Regensburg

Vol. 132. J.D. Westwood, R.S. Haluck, H.M. Hoffman, G.T. Mogel, R. Phillips, R.A. Robb and K.G. Vosburgh (Eds.), Medicine Meets Virtual Reality 16 – parallel, combinatorial, convergent: NextMed by Design

Vol. 131. R. Latifi (Ed.), Current Principles and Practices of Telemedicine and e-Health

Vol. 130. J.I. Westbrook, E.W. Coiera, J.L. Callen and J. Aarts (Eds.), Information Technology in Health Care 2007 – Proceedings of the 3rd International Conference on Information Technology in Health Care: Socio-technical Approaches

ISSN 0926-9630

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends

Edited by

Annette ten Teije

Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

Silvia Miksch

Danube University Krems, Krems, Austria

and

Peter Lucas

Radboud University Nijmegen, Nijmegen, The Netherlands

IOS
Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

ISBN 978-1-58603-873-1
Library of Congress Control Number: 2008931369

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the UK and Ireland
Gazelle Books Services Ltd.
White Cross Mills
Hightown
Lancaster LA1 4XS
United Kingdom
fax: +44 1524 63232
e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

LEGAL NOTICE
The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

A. Ten Teije et al. (Eds.)
IOS Press, 2008

Preface

This book, “Computer-based Clinical Guidelines and Protocols: a Primer and Current Trends”, is the result of the effort of the editors, started in 2006 with the organisation of an ECAI-2006 workshop at Riva del Garda titled “AI Techniques in Health Care: Evidence-based Guidelines and Protocols” and, subsequently, with the organisation of a workshop on “Computer-based Clinical Guidelines and Protocols (CCG’08)” at the Lorentz Centre of Leiden University at the beginning of 2008, to bring together researchers from the area of computer-based clinical guidelines and protocols with the aim of informing both researchers and others interested in clinical guidelines and protocols about the state of the art in this area. The ECAI-2006 workshop was a follow-up workshop from the “First European Workshop on Computerized Guidelines and Protocols” held in Leipzig, Germany in 2000 and the “Symposium on Computerized Guidelines and Protocols (CGP-2004)” held in Prague, Czech Republic in 2004.

With the current rise in the complexity and costs of health care, on the one hand, and increasing expectations of society about what health care is able to deliver, on the other hand, health-care professionals have developed a, sometimes urgent, need for care-practice support. Clinical guidelines and protocols have become the main instruments for disseminating best practices in health care. A clinical guideline gives general, usually nation wide, recommendations and instructions to assist the medical professional and the patient in decision making. In this book protocols are defined as local, specialised versions of guidelines, obtained in most cased by summarising information extracted from a guideline and by adding more detail, for example with regard to actual drugs or doses of drugs to be prescribed. As the detailed information may vary from hospital to hospital, the clinical protocols will reflect these differences between health care organisations.

Clinical guidelines and protocols promote safe practices, reduce inter-clinician practice variations and support decision-making in patient care while constraining the costs of care. In many cases, clinical guidelines and protocols have been useful in improving the quality and consistency of health care, by supporting health-care quality assessment and assurance, clinical decision making, work-flow and resource management. The benefits of having access to clinical guidelines and protocols are widely recognised, yet the guideline development process is time- and resource-consuming. In addition, the size and complexity of guidelines remains a major hurdle for effectively using them in clinical practice. Despite this, the number of clinical guidelines being developed and revised by professional health-care organisation has been rising steadily.

At the time when this preface was written, clinical guidelines were still textual documents, available in the form of booklets; it is only recent that these booklets have also become available in electronic form on the guideline-developers’ world-wide web sites. Thus, present-day guidelines are still far removed from being ‘computer-based’. With the now almost ubiquitous presence of information technology in modern society it is likely that this will change, and that clinical guidelines will become computer-based in the very near future. This development was already foreseen by a small number of researchers, who started doing research in computer-based guidelines more than a decade ago.

It has taken a relatively long period of time in comparison to other areas, such as banking, before computers were accepted as valuable tools by medical doctors and nurses for the clinical management of disease of patients. Many countries are now on the brink of the wide-scale introduction of electronic patient records, which implies that, after many centuries, paper will no longer be used to store patient information and that computers will even become more important than they already are in health care. In this context, it seems even more likely that clinical guidelines will become computer-based, i.e., computer interpretable and executable. However, in order to make this happen, there is still a large gap between the current practice of guidelines development, on the one hand, and computer-based guidelines, on the other hand, that needs to be bridged. This issue is addressed by some of the chapters in this book.

Many researchers expect that the computer-based development, use and dissemination of guidelines will have a positive effect on the time required for the development of new guidelines and protocols, for the revision of existing ones, for deployment in daily care and dissemination. Furthermore, computer-based methods are indispensable for ensuring that guidelines are in agreement with the latest requirement for guideline development.

This book brings together results from different branches of computer science (in particular, artificial intelligence), medical informatics and medicine to examine cutting-edge approaches to computer-based guideline modelling, verification and interpretation. Different methods have been developed to support the development, deployment, maintenance and use of evidence-based guidelines, using techniques from artificial intelligence, software engineering, medical informatics and formal methods. Such methods employ different representation formalisms and computational techniques. As the guideline-related research spans a wide range of research communities, a comprehensive integration of the results of these communities was lacking. It is the intention of the publication of this book to fill this gap. It is the first book of its kind that partially has the nature of a textbook.

The book consists of two parts. The first part consists of 9 chapters which together offer a comprehensive overview of the most important medical and computer-science aspects of clinical guidelines and protocols. Not only are these chapters meant as a review of the state of the art, since, in addition, these chapters indicate cross links between topics and directions for future research. All chapters were written by authors with extensive expertise in the covered areas. Topics covered are: guideline development and deployment in medical practice, guideline representation languages, guideline modelling methods, use of formal methods in guideline development, temporal aspects of guidelines, planning, guideline adaptation, visualisation of guidelines and guideline compliance.

The second part of the book consists of chapters that are extended versions of selected papers that were originally submitted to the ECAI-2006 workshop mentioned at the beginning of this preface. These chapters will provide the reader detailed information about actual research in the area by leading researchers.

Chapters in both parts of the book have been extensively reviewed and profited from the feedback received in the writing process.

Thanks should go to the people–unfortunately too many to explicitly mention here–who helped in reviewing the various chapters included in the book and who provided very useful feedback to the authors. Finally, we are grateful to the Lorentz Centre at Leiden University for the facilities they offered in the process of completing the book, without which it would not have been possible to achieve the level of quality we were able to reach.

The Editors, 14th March, 2008,

Annette ten Teije,
Amsterdam

Silvia Miksch,
Krems

Peter J.F. Lucas,
Nijmegen

This page intentionally left blank

Contents

Part I. A Primer

Part II. Current Trends

Part I

A Primer

This page intentionally left blank

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-3

Chapter 1
Guideline Development

Kitty ROSENBRAND[1], Joyce VAN CROONENBORG, Jolanda WITTENBERG
Dutch Institute for Healthcare Improvement CBO, Utrecht, the Netherlands

Abstract During the last decade many countries have become increasingly interested in the development and use of evidence-based practice guidelines, recognising that guidelines are key tools to improve the quality and appropriateness of health care. They are considered to be the ideal mediator for bridging the gap between the growing stream of research findings and actual clinical practice. Systematic reviews of guideline evaluations have shown that clinical practice guidelines can be an effective means of both changing the process of healthcare delivery and improving outcomes. A review of 59 guideline evaluation studies found that, in all but 4, statistically significant improvements occurred in clinical practice after implementation [17]. A systematic review of 87 studies on the use of guidelines concluded that 81 studies revealed evidence of improved patient outcomes [12].

Evidence-based guidelines are becoming an important and indispensable part of quality healthcare because of their potentials to improve quality and also reduce cost of health-care. Adherence to guidelines and protocols may reduce health-care costs up to a 25% [11]. We will present an overview of the history of guideline-development and give some widely used definitions of guidelines. Guidelines are developed in a structured and systematic way, this process will be explained later. Also implementation tools necessary to put the guidelines into practice in an active way, will be discussed.

Keywords: guideline development process, evidence-based guidelines, clinical indicators, quality assesment , guideline implementation, living-guidelines

Introduction

In 1977 the National Institute of Health (US) started with a consensus development Program. Two years later the Canadian Task Force on the Periodic Health Examination (now Canadian Task Force on Preventive Health Care) added the first “levels of evidence” [9].

The Dutch Institute for Healthcare Improvement CBO started with consensus guideline development in 1980. The 1st guideline on blood transfusion was published in 1982. The Dutch College of General Practitioners has been developing primary care guidelines since 1989. The Agency for Health Care Policy and Research (AHCPR, now Agency for Health Research and Quality, AHRQ) started their National evidence-based guideline program in 1989 until 1996. From 1996 on they write evidence reports, which are the scientific basis for evidence based guideline development. Since the mid

[1] Corresponding Author: Kitty Rosenbrand, Dutch Institute for Healthcare Improvement CBO, Churchillaan 11, 3527GV, Utrecht, The Netherlands; E-mail: k.rosenbrand@CBO.NL

nineties many organizations world-wide started guideline development programs, e.g. the Scottish Intercollegiate Guidelines Network (SIGN), the National Institute for Clinical Excellence (NICE), New Zealand Guidelines Group (NZGG), Guidelines Advisory Committee (GAC) Canada, Current Care / Duodecim - Finnish Medical Society, National Federation of Cancer Centres (FNCLCC) in France.

Traditionally, guidelines have been based on consensus amongst experts. However, this process has its limitations, it usually only includes some but not all perspectives and can lead to flawed conclusions because expert opinion does not always reflect the state of current knowledge [2]. Furthermore, it is necessary for research literature to be analyzed systematically in order to avoid biased conclusions [34]. It is now widely accepted that guideline recommendations should be based on systematic identification and synthesis of the best available scientific evidence. This may be a daunting task given the size of research activity in some clinical areas. Next to this trend of consensus to evidence based guidelines we see more changing trends in time (see table 1), e.g. from monodisciplinary to multidisciplinary guidelines, focus from development to implementation of guidelines etc.

From	**To**
regional guidelines from professional groups	national guideline programmes
informal consensus	evidence-based
monodisciplinary	multidisciplinary
focus on development	focus on implementation
limited life-expectancy	‘living guidelines’
paper versions	Internet
guidelines for clinicians	patient versions and patient involvement

Table 1. Trends in guideline development

1. Definitions Guidelines and Protocols

There is some disagreement over which documents should be called guidelines; the term often being used interchangeably with protocols when implying a greater degree of compliance [35]. It has been suggested that the term ‘guideline’ be applied only to a systematically developed advisory statement devised according to validated scientific methodologies [33].

There are several definitions for guidelines and protocols. Clinical practice guidelines have been defined as decision tools to close gaps between current and optimal practice [26], but they are also described as:

- mechanisms to improve the quality of health care and decrease costs and utilisation [4]
- recommendations devised to influence decisions about health interventions [5]
- tools to outline procedures to be followed thus helping doctors make decisions [30]
- processes to operationalise the implementation of evidence-based practice [33].

A widely used definition of guidelines is that of the Institute of Medicine (IOM): *'Guidelines are systematically developed statements to assist practitioner and patient decisions about appropriate healthcare for specific clinical circumstances'.* [15].

CBO has specified the following so called 'Haamstede' definition [14]: *'A guideline is a document with recommendations and instructions to assist the medical professional and the patient in decision making, based on results of scientific research followed by discussion and expression of expert-opinions, to make effective and efficient medical practice explicit.'*

The NZ Guidelines Group (NZGG) has a broad definition of clinical practice guidelines: *'Guidelines provide guidance in decision making at each level of interaction; between health professional and consumer, between purchaser and provider, and between 'funder' and 'purchaser''*

NZGG defines also different types of guidelines (http://www.nzgg.org.nz).

Consensus Based Guideline: *The most common form of guideline developed is agreement among a group of experts.*

Evidence Based Guideline: *Developed after the systematic retrieval and appraisal of information from the literature. "They usually include strategies for describing the strength of the evidence, and try to clearly separate opinions from evidence ...they make statements not just about which of two treatment options is 'better', but quantify the absolute differences in outcome, including both benefits and harms".*

Explicit Evidence Based Guideline: *Developed as an evidence based guideline, "...but also projects the healthcare outcomes (benefits, harms, utilization and costs) of the change in practice on a defined population".*

Guidelines that have recommendations that are based on evidence are considered to be of greater value to practitioners and consumers because the decisions are likely to result in improved consumer outcomes [19].

2. Aims of Guidelines

Guidelines are developed to summarise and synthesize knowledge and innovations in medicine, to reduce variation in practice, promote evidence-based clinical practice and satisfy the need for transparency and accountability. The ultimate goal of guidelines is to improve the quality of patient health care [10]. However, guidelines are necessarily general and there will be circumstances when their recommendations are not appropriate for an individual patient. Healthcare professionals are expected to take clinical guidelines fully into account when exercising their clinical judgement. The guidance does not, however, override the responsibility of healthcare professionals to make decisions appropriate to the circumstances of each patient, in consultation with the patient and/or guardian or carer. Healthcare professionals should document the reasons for not following a guideline.

3. Guidelines International Network (G-I-N)

The Guidelines International Network (www.g-i-n.net) is a major international initiative involving guideline-developing organisations from around the world. G-I-N seeks to improve the quality of health care by promoting systematic development of clinical practice guidelines and their application into practice.

Over 70 organisations from 35 countries joined the Guidelines International Network G-I-N, including national institutions from Oceania, North and South America, Europe, and the WHO. Most GIN-members prepare evidence based clinical practice guidelines, or actively promote the use of evidence in practice. One of the priorities of the organisation is to share evidence tables and adapt guidelines for local circumstances based on international evidence.

Although many countries have built up experience in the development, appraisal, and implementation of guidelines, until G-I-N was founded there was no established forum for collaboration at an international level. As a result, in different countries seeking similar goals and using similar strategies, efforts have been unnecessarily duplicated and opportunities for harmonisation lost because of the lack of a supporting organisational framework. A baseline survey confirmed a strong demand for such an entity. A multinational group of guideline experts initiated the development of a non-profit organisation aimed at promotion of systematic guideline development and implementation. An international guidelines forum to promote information sharing and cooperation was proposed in 2001, building on existing partnerships in the guidelines field. A multinational group initiated the foundation of the network and as a result the Guidelines International Network (G-I-N) was founded in November 2002. One year later the Network released the International Guideline Library, a searchable database that now contains more than 4000 guideline resources including published guidelines, guidelines under development, "guidelines for guidelines", training materials, and patient information tools [36].

Examples of G-I-N members include:
The Scottish Intercollegiate Guidelines Network (SIGN): www.sign.ac.uk
National Institute for Health and Clinical Excellence (NICE): www.nice.org.uk
The New Zealand Guidelines Group (NZGG): http://www.nzgg.org.nz
The Dutch Institute for health care improvement (CBO): www.cbo.nl (in Dutch)

4. Dutch Institute for Healthcare Improvement (CBO)

The Dutch Institute for Healthcare Improvement CBO, founded in 1979, is a not-for-profit, national knowledge-, innovation- and implementation-institute that advises, supports and trains healthcare providers (professionals, hospitals) encouraging their collaboration aimed at achieving breakthrough results in the improvement of the quality of patient care. The mission of CBO is to make a significant contribution to the improvement of patient care in the Netherlands.

The aim of CBO's guideline development program is to contribute to the quality improvement, effectiveness and efficiency of clinical care for patients by changing practice based on high quality information. This aim is realised by developing evidence based national guidelines, which can be translated into protocols for local practice. Guidelines are developed under auspices of the Medical Scientific Board of the CBO, in close co-operation with the Order of Medical Specialists and the Scientific Medical Societies.

5. Guideline Development Process of CBO

All members of G-I-N have their own methodology of evidence based guideline development. Although they all embrace the same principles (i.e. systematic literature search, evidence appraisal). See for an inventory on recent guidelines manuals the G-I-N website [2].

In the next paragraphs the methodology of evidence-based guideline development of the Dutch Institute for Healthcare Improvement (CBO) will be described. The methodology of other institutes can slightly differ. The authors thank the Scottish Intercollegiate Guidelines Network (SIGN) as major text passages are derived from their (English) guideline development handbook[3].

5.1. Composition of the Guideline Development Group

The process of developing guidelines should include participation by representatives of key groups and disciplines affected. Clinical practice guidelines should be developed by physicians in collaboration with representatives of those who will be affected by the specific intervention(s) in question, including relevant physician groups, patients, and other health care providers as appropriate.

Establishing a multidisciplinary guideline development group is therefore important to ensure that:

1. all relevant groups are represented, providing expertise from all stages in the patient's journey of care
2. all relevant scientific evidence will be located and critically evaluated
3. practical problems with using the guideline will be identified and addressed

[2] www.g-i-n.net/index.cfm?fuseaction=membersarea&fusesubaction=article&documentid=64&articleID=170

[3] http://www.sign.ac.uk/guidelines/fulltext/50/index.html

4. stakeholder groups will see the guideline as credible and will cooperate in implementation.

CBO guideline development groups vary in size depending on the scope of the topic under consideration, but generally comprise between 15 and 25 members. Care is also taken to ensure that the group is balanced geographically, with representatives from across the Netherlands.

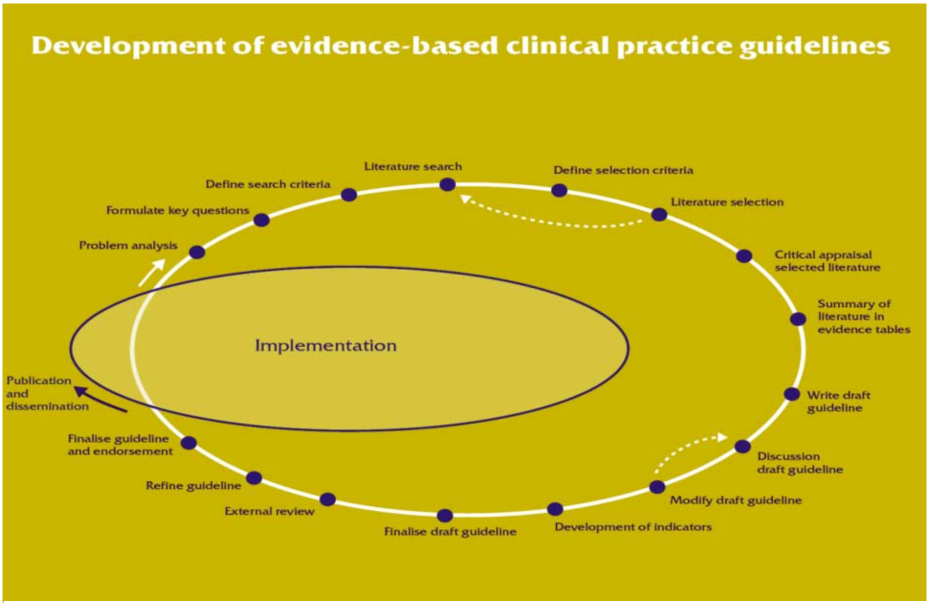

Development of evidence-based clinical practice guidelines
Literature search
Define selection criteria
Define search criteria
Literature selection
Formulate key questions
Problem analysis
Critical appraisal selected literature
Summary of literature in evidence tables
Implementation
Publication and dissemination
Write draft guideline
Finalise guideline and endorsement
Discussion draft guideline
Refine guideline
Modify draft guideline
External review
Finalise draft guideline
Development of indicators

Figure 1: CBO Guideline Development Cycle

5.2. Problem Analysis

For a schematic overview of CBO's guideline development methodology see figure 1. After the guideline development group has been initialised the group starts to discuss the items the guideline should address. This so called 'problem analysis' consists of interviews or questionnaires with stakeholders, disciplines affected and patient groups. Results of the analysis consist of medical, organisational and patient issues to address in the guideline to be developed.

5.3. Formulation of Key Questions

The training in critical appraisal and guideline development offered to members of CBO guideline development groups encourages them to break down the guideline remit into a series of structured key questions that clearly identify the population concerned, the intervention (or diagnostic test, etc.) under investigation, the type of control used, and the outcome measures used to measure the effectiveness of the interventions. These questions then form the basis of the literature search, which is undertaken by a CBO information- specialist.

5.4. Literature Search

The search must focus on the best available evidence to address each key question, and should ensure maximum coverage of studies at the top of the hierarchy of study types*. Study types include:

1. Meta-analyses, and systematic reviews*
2. Randomised controlled trials*
3. Observational studies
4. Diagnostic studies
5. Economic studies
6. Qualitative studies
7. Guidelines

In order to minimise bias and to ensure adequate coverage of the relevant literature, the literature search must cover a range of sources. Sources can include: the Cochrane Library, Medline, Embase, the Internet, etc.

5.5. Critical Appraisal of Selected Literature

Once papers have been selected as potential sources of evidence, the methodology used in each study is assessed to ensure its validity. The result of this assessment will affect the level of evidence allocated to the paper, which will in turn influence the grade of the conclusion that it supports (see later this paragraph).

The methodological assessment of studies in the guideline is based on a number of questions that focus on those aspects of the study design that research has shown to have a significant influence on the validity of the results reported and conclusions drawn. These questions differ between study types. A range of checklists is used to bring a degree of consistency to the assessment process. Checklists are available at the website of CBO .[4]

5.6. Summary of Literature in Evidence Tables

Evidence tables are compiled based on the quality assessments of individual studies. The tables summarise all the validated studies identified from the systematic literature review relating to each key question. They are presented in a standard format to make it easier to compare results across studies, and will present the evidence for each outcome measure used in the published studies. These evidence tables form an essential part of the guideline development record and ensure that the basis of the guideline development group's recommendations is transparent.

[4] http://www.cbo.nl/product/richtlijnen/handleiding_ebro/article20050427141202/view

5.7. Writing Draft Guideline

To make the guideline development steps transparent to the reader CBO uses a standard format for the guidelines. This format includes the following elements:

- Key question to be answered
- Summary of the evidence, preferably using
- Evidence tables
- Conclusion(s) including level of evidence
- Other considerations, e.g.
  - Clinical relevance and safety
  - Patient's perspective
  - Organisational consequences
- Recommendation(s)
- References

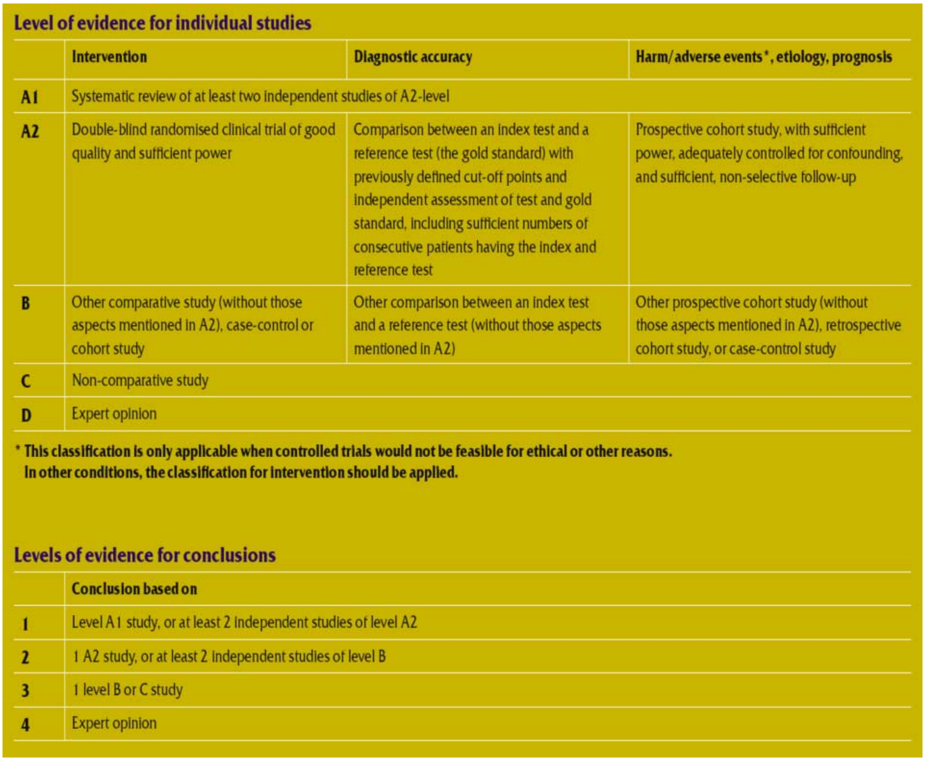
5.8. Grading of the Evidence

Clinical guidelines are only as good as the evidence and judgments they are based on. Not all evidence used in a guideline is of the same strength and quality. To make this transparent the evidence is graded. Since the 1970s a growing number of organisations have employed various systems to grade the quality (level) of evidence and the strength of recommendations. Unfortunately, different organisations use different systems to grade the quality of evidence and the strength of recommendations [23, 24, 25].

In 2000 an informal collaboration of people with an interest in addressing the shortcomings of present grading systems in health care started the Grading of Recommendations Assessment, Development and Evaluation (short GRADE) Working Group. Their aim is to develop a common, sensible approach to grading quality of evidence and strength of recommendation [3].

At CBO individual studies and conclusions (summary statements) based on those studies are graded according to the system below (see figure 2). The grading of individual studies is based on study design and methodological quality. Grading of the conclusions, includes the number and type of individual studies that support the conclusion. CBO only grades the evidence from literature, not the recommendation. The recommendation is the result of balancing the evidence and other considerations, such as patient views and applicability. Some other guideline developers do not use the method of summary statements of the evidence, they grade the recommendations.

It is important to emphasise that the grading does not relate to the importance of the recommendation, but to the strength of the supporting evidence and, in particular, to the predictive power of the study designs from which that data was obtained. Thus, the grading assigned indicates to users the likelihood that, if that recommendation is implemented, the predicted outcome will be achieved.



Level of evidence for individual studies

	Intervention	Diagnostic accuracy	Harm/adverse events*, etiology, prognosis
A1	Systematic review of at least two independent studies of A2-level		
A2	Double-blind randomised clinical trial of good quality and sufficient power	Comparison between an index test and a reference test (the gold standard) with previously defined cut-off points and independent assessment of test and gold standard, including sufficient numbers of consecutive patients having the index and reference test	Prospective cohort study, with sufficient power, adequately controlled for confounding, and sufficient, non-selective follow-up
B	Other comparative study (without those aspects mentioned in A2), case-control or cohort study	Other comparison between an index test and a reference test (without those aspects mentioned in A2)	Other prospective cohort study (without those aspects mentioned in A2), retrospective cohort study, or case-control study
C	Non-comparative study		
D	Expert opinion		

* This classification is only applicable when controlled trials would not be feasible for ethical or other reasons. In other conditions, the classification for intervention should be applied.

Levels of evidence for conclusions

	Conclusion based on
1	Level A1 study, or at least 2 independent studies of level A2
2	1 A2 study, or at least 2 independent studies of level B
3	1 level B or C study
4	Expert opinion

Figure 2: CBO level of evidence

5.9. Other Considerations

It is rare for the evidence to show clearly and unambiguously what course of action should be recommended for any given question. Consequently, it is not always clear to those who were not involved in the decision making process how guideline developers were able to arrive at their recommendations, given the evidence they had to base them on. In order to address this problem, CBO has introduced the concept of other considerations [41]. Under the heading of other considerations, guideline development groups summarise their view of the evidence in relation to clinical relevance, clinical impact, safety, patient's perspective, generalisability of study findings, organisational and implementation consequences.

5.10. External Review

All CBO guidelines are reviewed in draft form by independent expert referees. There are several ways to organise the external review:

1. Review of draft guidelines by scientific medical societies (paper or internet)
2. National open meeting

CBO organises the external review to discuss the draft recommendations of each guideline. This takes place whilst the guideline is still in development and gives the guideline development group the opportunity to present their preliminary conclusions and draft recommendations to a wider audience. The benefits are twofold:

1. the guideline development group obtain valuable feedback and suggestions for additional evidence which they might consider, or alternative interpretation of that evidence
2. the participants are able to contribute to and influence the form of the final guideline, generating a sense of ownership over the guideline across geographical and disciplinary boundaries.

5.11. Guideline Endorsement and Publication

After the external review phase the guideline is finalised. The final guideline is sent to all participating medical scientific societies to sign for the official endorsement of the guideline.

All CBO guidelines and summaries of the guideline, along with any updates to guidelines, are available free of charge on the CBO website: www.cbo.nl. In the past also a printed edition of the guideline was published, but this is not common to date.

6. Examples of Guidelines

In this section, we will discuss two case studies that give an impression about the structure of guidelines. First, we will discuss the guideline on the treatment of diabetes mellitus type 2 (DM2). Then, the guideline on the treatment of breast cancer is discussed. These guidelines are considerably distinct as the diabetes guideline is aimed at the general practitioner, whereas the breast cancer guideline is developed for medical specialists. As a result, the latter is more extensive in its justifications, whereas the diabetes guideline contains more detail.

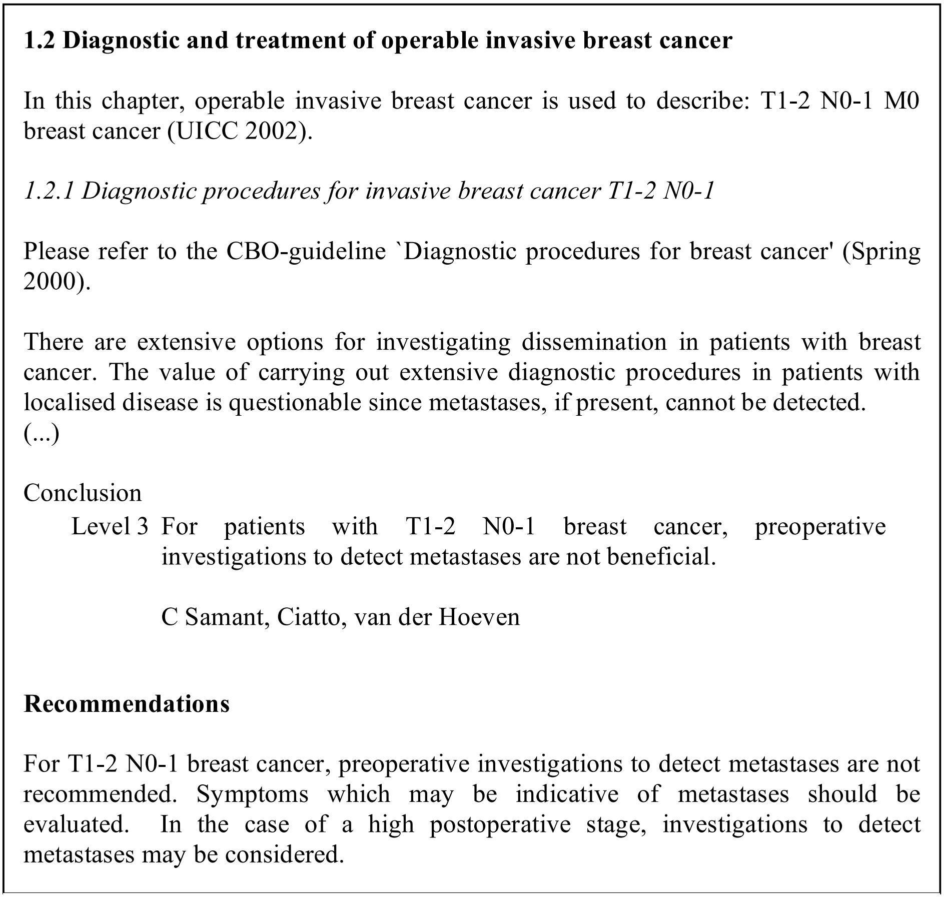
6.1. Diabetes Mellitus Type 2

In 2003, about 36 per 1000 men and 39 per 1000 women were diagnosed with diabetes mellitus type 2 in the Netherlands. Worldwide, the prevalence of diabetes is rising due to population growth, aging, urbanisation, and increasing prevalence of obesity, and physical inactivity. An example of a part of a guideline is the following (translated) text:

- refer to a dietician; check blood glucose after 3 months
- in case (1) fails and Quetelet Index (QI) <= 27, then administer a sulfonylureum derivate (e.g., tolbutamide, 500 mg 1 time per day, max. 1000 mg 2 per day) and in case of Quetelet Index (QI) > 27 biguanide (500 mg 1 per day, max. 1000 mg 3 times per day); start with lowest dosage, increase each 2-4 weeks if necessary.

This guideline is particularly concise (about 3 A4 pages). While modern guidelines can be as large as 100 pages, the number of recommendations they include are typically few. In complicated diseases, each type of disease is typically described in different sections of a guideline, which provides ways to modularise the guideline in a natural fashion.

6.2. Breast Cancer



1.2 Diagnostic and treatment of operable invasive breast cancer

In this chapter, operable invasive breast cancer is used to describe: T1-2 N0-1 M0 breast cancer (UICC 2002).

1.2.1 Diagnostic procedures for invasive breast cancer T1-2 N0-1

Please refer to the CBO-guideline `Diagnostic procedures for breast cancer' (Spring 2000).

There are extensive options for investigating dissemination in patients with breast cancer. The value of carrying out extensive diagnostic procedures in patients with localised disease is questionable since metastases, if present, cannot be detected.
(...)

Conclusion

Level 3 For patients with T1-2 N0-1 breast cancer, preoperative investigations to detect metastases are not beneficial.

C Samant, Ciatto, van der Hoeven

Recommendations

For T1-2 N0-1 breast cancer, preoperative investigations to detect metastases are not recommended. Symptoms which may be indicative of metastases should be evaluated. In the case of a high postoperative stage, investigations to detect metastases may be considered.

Figure 3: Fragment of the breast cancer guideline

In the Netherlands only, as many as 10,000 women are diagnosed with breast cancer every year. For women, the chance of ever being diagnosed with this disease is 10%. Changes in DNA, in particular the genes that control the instructions for cells to grow, divide, and die, may cause cancer; however, little is known under which circumstances this actually happens.

The guideline that we discuss here is the 2004 version of the Dutch CBO guideline on the treatment of breast cancer. This guideline is considerably more complex than the diabetes guideline due to the fact that it was developed more systematically described by the methodology in the previous section. The guideline is divided in several chapters. The first chapter contains introduction; the other chapters have a specific topic related to the primary topic which do not overlap with other chapters. All chapters are divided in subsections that contain:

a. *summary text,* which normally serves as an introduction to the issues that follow so that the reader is able to understand the arguments underlying recommendations and conclusions;
b. *conclusions*: these are short summary statements of the important insight from the literature, introduced in the preceding guideline text.
c. *recommendations*: these are statements pertaining to (medical) management actions.

See Figure 3 with a fragment from this guideline.

The structure of these chapters obviously depends on the questions they want to answer as described by the methodology. For each question the primary literature is listed together with additional considerations. From the primary literature the most important conclusions are given a `grade of evidence' and are put in a separate box, which is based on the level of evidence of the individual studies. Finally, note that the recommendations follow from the primary literature and the additional considerations, i.e., not merely from the conclusions that are highlighted.

7. Implementation of Guidelines

There is no single answer to what is a successful implementation strategy although the limited research carried out suggests a range of approaches is more likely to succeed than a single approach. Ideally, the research literature should guide this phase of guideline development but methodological limitations of the research base mean that this is not necessarily possible. For example, problems of sample sizes, length of time required before data analysis can begin, resources issues, and different health systems often mean that their search cannot be transferred or generalised to other settings. Numerous theories explain behaviour change and support the use of different interventions to bring about modifications in practice. Which ones are as yet the most effective and efficient is unclear [18]. Instead a number of different theoretical approaches contribute to our understanding of the process of change. For a more detailed summary of the change theories see [22].

To change behaviour is possible, but this change generally requires comprehensive approaches at different levels (doctor, team practice, hospital, wider environment), tailored to specific settings and target groups [20].

Levels of implementation change that should be considered are:

- The practitioner – patient level e.g. changing clinician/patient behaviour and attitudes
- At the systems level e.g. enabling clinicians to make changes easily by providing access to computer decision support systems
- At the policy level e.g. by providing coverage decisions that enable access to health interventions

Decision support systems include anything manual or automated that prompt health professionals to perform a clinical action. Examples are reminders about screening, laboratory reports where results to note are highlighted, follow up appointment systems and stickers on charts. In particular, computerised decision support systems have led to improvements in doctors' decision making on drug dosage, provision of preventive care and general clinical management of patients [27]. The advantage of these systems is that they are fairly easy to implement and are available to clinicians at the time required (so called 'just in time reminders', information is prompt available at the moment the clinician is asking for it).

To achieve the objective "to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" [15]. it is important not only to develop valid guidelines by a sound methodology, but also to ensure the implementation of the evidence-based recommendations. As one of a range of tools to help health care professionals and organisations to improve clinical effectiveness and patient outcomes, guidelines provide an opportunity for practitioners to improve shared clinical decision-making, increase team working, expand their evidence-based knowledge, and reduce variation in practice. They can also enable professionals to keep up to date and to assess their own clinical performance against the recommendations for best practice. However, there is often a gap between the development of guidelines, as set out in the previous sections, and their implementation into practice. Just as guidelines themselves help provide a bridge between research and practice, this paragraph outlines the strategies that can assist practitioners, and health services to bridge the gap between guideline development and implementation.

7.1. The GuideLine Implementability Appraisal (GLIA)

GLIA is a tool for appraisal of implementability of clinical guidelines, an instrument designed to identify any potential obstacles to guideline implementation [39]. GLIA may be useful to guideline developers who can apply the results to remedy defects in their guidelines. Likewise, guideline implementers may use GLIA to select implementable recommendations and to devise implementation strategies that address identified barriers. By aiding the design and operationalisation of highly implementable guidelines, application of GLIA may help to improve health outcomes, but further evaluation will be required to support this potential benefit.

8. Clinical Indicators

Clinical indicators give an indication of the quality of the patient care delivered. In most health care systems, a consensus is emerging that there is a need for quality measures. Various audiences may wish to use them to document the quality of care, make comparisons (benchmarking), make judgments and determine priorities, support accountability, support quality improvement, and provide transparency in health care [40, 37]. Using clinical indicators is one way of measuring and monitoring the quality of care and services.

8.1. Types of Indicators

Indicators can be divided in structure, process and outcome indicators. Structure indicators give information of the (organisational) limiting conditions in which health care is delivered. Examples of structure indicators are ‘percentage of teams for diabetes care including a foot therapist’ or ‘existence of a stroke unit’.

Process indicators give information on actions performed in care processes. Process indicators can be influenced directly; they measure how (often) something is done. An example of this type of indicators is ‘percentage of diabetes patients getting an annual eye test’

Outcome indicators give information on the outcome of care processes measured at patient level. They depend on many factors and therefore are difficult to reduce to actual patient care. An example of an outcome indicator is ‘ percentage of patients with severe pain at 36 hours after surgery’.

8.2. Guidelines and Indicators

A well-founded judgement of quality of a specified care process is only possible in a validated way (by indicators) by measurement of the quality criteria as described in an (multidisciplinary) evidence-based guideline, authorised by medical scientific societies. Ideally indicators are based on guidelines. However a good evidence based guideline is not always available. In such cases indicators are based on the best available evidence about the quality of care. Therefore indicators can be a starting point for the formulation of a new guideline. Next to this the data collected by measurement with indicators can give impulse to adjustment or actualisation of a guideline (living guidelines).

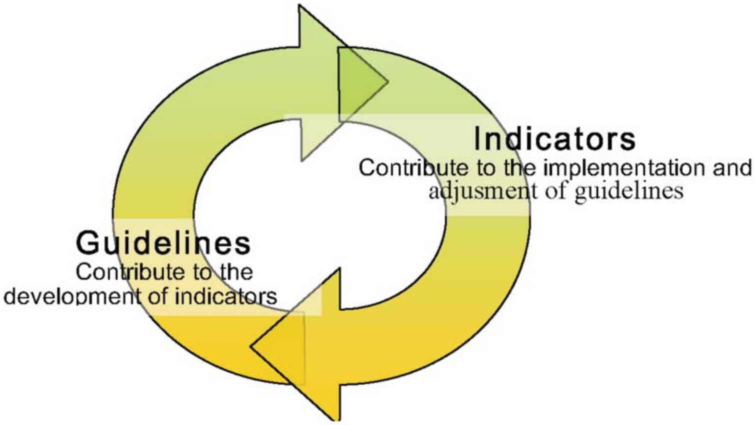

Indicators
Contribute to the implementation and adjusment of guidelines
Guidelines
Contribute to the development of indicators

Figure 4: Relation between guidelines and clinical indicators

8.3. Use of Clinical Indicators

Traditionally indicators can be divided in “internal” and “external” indicators. The goals of internal indicators is monitoring and improving care processes or professional performance within the own organisation. By measuring performance on a continue basis healthcare delivery can be examined critically or developments can be followed (for example introduction of a protocol).

External indicators are used for being accountable to government institutions, health insurance companies or consumers about the quality of care.

A good data registration is necessary for all types of indicators. Ideally data are directly registered within the care process itself. However it is still daily practice that data are often only registered in paper documents and rarely in a digital way. This costs a lot of time and money. The collection of data for indicators is therefore often a separate activity out of the primary care process (and therefore expensive). Besides data registered in a digital way are requiring also effort to extract out of hospital systems [29, 42].

8.4. Development of Indicators

It is imperative that clinical indicators are meaningful, scientifically sound, generalisable, and interpretable. To achieve this, clinical indicators must be developed, tested, and implemented with scientific rigor.

Indicators are selected from research data with consideration for optimal patient care (preferably an evidence-based guideline), supplemented by expert opinion. In the selection procedure, the feasibility, such as their measurability and improvability, is important beside validity and reliability. A clinical indicator should be defined exactly and expressed as a quotient. After a try-out, the measurements and reporting should follow. The report contains an in-depth analysis of causal and contributing factors associated with the measured results. A description of the clinical circumstances and a correction for case mix should be included to allow for a justified interpretation. Initially, when evidence links a process to better outcomes it may appear that the standard for a proportion of patients so treated should be 100%. However, there are reasons why this is not always the case, depending on how well the denominator of eligible patients can be defined. The indicators must be part of an improvement strategy, for which comparison feedback is often used. Comparison with reference data can be used to construct improvement programs [29, 42].

9. Quality Assessment of Guidelines – AGREE Instrument

Although the principles for the development of sound evidence–based guidelines are well established, many published guidelines fall short of the internationally consented quality criteria for their production and use. In response several national and international initiatives have been working on programmes for the promotion of quality in guideline development and use.

Guidelines should meet specific quality criteria to ensure good quality. Users should be able to be confident that potential biases inherent of guideline development have been addressed appropriately and that the recommendations for practice are both internally and externally valid as well as feasible for practice [1]. However, recent studies have reported that the methodological quality of many guidelines is modest and is heterogeneous between the different guidelines and different guideline programs [7, 16, 28, 38]. Although clinical guidelines can provide a solution to some of the important problems in patient care, there are issues that need to be tackled before guidelines can achieve their full potential [21]. A set of criteria for high quality guidelines was developed and validated by an international group of researchers and guideline developers (the AGREE collaboration). Some cancer guidelines (including those produced by the French National Federation of Cancer Centres – FNCLCC the SOR) were used in the validation process for these criteria. Recommendations for guideline developers will help researchers and practitioners in health care to develop high quality guidelines for the management of their patients.

10. Research Agenda

Guideline developers must find a way to effectively communicate and work with IT scientists to develop standards and protocols for the translation of (trans-) national guidelines into electronic formats. In the longer term, so called “living guidelines” that can be continuously updated and used by a number of different countries will be a great advancement.

To be effective, there must be formal internationally agreed standards that allow electronic guidelines to be shared and automatically updated [13]. To succeed guideline developers need to work in close cooperation with designers and vendors of electronic decision support systems and tools [6].

10.1. Living Guidelines

In the current scenario of guideline development, dissemination and deployment, there is a major problem with clinical guidelines:

Recommendations can be outdated or not applicable in practice, because most guidelines are only revised every 5 years. In contrast with this, scientific and pragmatic knowledge is growing faster every year. At this moment, a guideline is a static document, which cannot be modified easily. This problem has led to a future challenge, often referred to as "living guidelines": Update of the guidelines on a more continuous basis: clinical guidelines have to become flexible, adaptable documents. The aim is to develop guidelines, which present up-to-date and state-of-the-art knowledge to practitioners.

To make this possible, guidelines have to be modular in structure, so that only part of a guideline can be adjusted and not the whole document needs revision. To make the approach of living guidelines possible, there must be some major changes in the guideline development process. Most guidelines are authored in an unstructured narrative form. Computer-based support depends on a more formal, structured representation, and can be used to address a number of challenges at all stages of the guideline life-cycle: modelling, authoring, dissemination, implementation and update (see figure 5). At this moment, guidelines are often multi-interpretable [31]. Also, different guidelines can include the same modules, which can be in conflict with other contents of the guidelines. Guidelines are complex documents. As a result, guidelines can be ambiguous, incomplete and even inconsistent [32]. In the modelling phase of guidelines, methods have to be developed to support this process. Also, terminology is a problem here; precise, abstract definitions of core notions for medical management are necessary.

In summary: to enable living guidelines, they must be developed in a more structured way. Formal methods can be of help here. This will be an important first step to enhance further computer-based support of guidelines and protocols.

The Protocure project presents our experience with applying formal methods to medical guidelines.

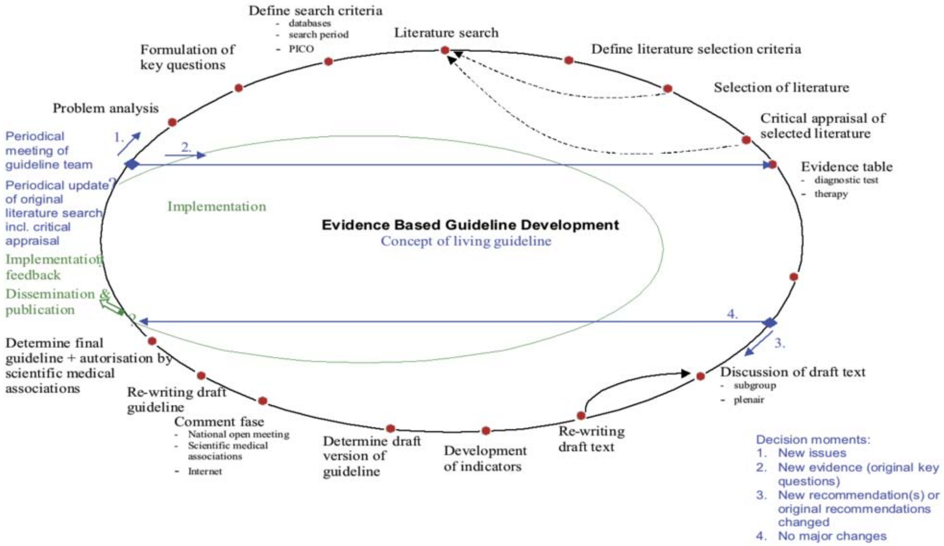

Define search criteria
- databases
- search period
- PICO
Literature search
Formulation of key questions
Define literature selection criteria
Selection of literature
Problem analysis
Critical appraisal of selected literature
Periodical meeting of guideline team
1.
2.
Evidence table
- diagnostic test
- therapy
Periodical update of original literature search incl. critical appraisal
Implementation
Evidence Based Guideline Development
Concept of living guideline
Implementation feedback
Dissemination & publication
4.
3.
Determine final guideline + autorisation by scientific medical associations
Re-writing draft guideline
Discussion of draft text
- subgroup
- plenair
Comment fase
- National open meeting
- Scientific medical associations
- Internet
Determine draft version of guideline
Development of indicators
Re-writing draft text
Decision moments:
1. New issues
2. New evidence (original key questions)
3. New recommendation(s) or original recommendations changed
4. No major changes

Figure 5. Living Guideline Cycle

Acknowledgements

We wish to thank Arjen Hommersom for his contribution in particular in section 6.

References

[1] AGREE Collaborative Group. Guideline development in Europe. *An international comparison. Int J Technol Assess Health Care 2000*; 16:1039-49.

[2] Antman K, Ayash L, Elics A et al. A phase II study of high dose cyclophosphamide,thiopeta and carboplatinum with autologous marrow support in women with measurable advanced breast cancer responding to standard dose therapy. *J Clin Oncol* 1992;10:102-10.

[3] Atkins D, Best D, Briss PA, et al . Grading quality of evidence and strength of recommendations. *BMJ* 2004; 328 (7454): 1490.

[4] Basinski A. Evaluation of clinical guidelines. *Can Med Assoc J* 1995; 153:1575 -1581.

[5] Berg M. Problems and promises of the protocol. *Soc Sci Med* 1997; 44:1081-1088.

[6] Biondich PG, Downs SM, Carroll AE, Shiffman RN, McDonald CJ. Collaboration between the medical informatics community and guideline authors: fostering HIT standard development that matters. *AMIA Annu Symp Proc 2006*;36-40.

[7] Burgers JS, Grol R, Klazinga NS, Mäkelä M, Zaat J; AGREE Collaboration. Towards evidence-based medicine: an international survey of 18 clinical guideline programs. *Int J Qual Health Care* 2003; 15(1):31-45.

[8] Burgers JS, van Everdingen JJ. Evidence-based guideline development in the Netherlands: the EBRO platform. *Ned Tijdschr Geneeskd* 2004;148(42):2057-9. (Dutch).

[9] Canadian Task Force on the Periodic Health Examination: The periodic health examination. *Can Med Assoc J* 1979; 121: 1193-1254.

[10] Carnett W. Clinical practice guidelines: a tool to improve care. *Qual Manag Health Care* 1999; 8:13-21.

[11] Clayton P, Hripsak G. Decision support in healthcare. *International Journal of Biomedical Computing* 1995; 39: 59-66

[12] Effective Health Care Bulletin. Implementing clinical practice guidelines. York: University of York, 1994.

[13] Elkin P, Peleg M, Lacson R, et al. Toward Standardization of Electronic Guideline Representation. *MD Computing* 2000;17(6):39-44.

[14] Everdingen J.J.E van et al. Evidence-based richtlijnontwikkeling. Een leidraad voor de praktijk. Houten: Bohn Stafleu van Loghem, 2004.

[15] Field MJ, Lohr KN. Clinical Practice Guidelines. Directions for a New Program. Institute of Medicine, Washington, DC: National Academy Press, 1990.

[16] Grilli R, Magrini N, Penna A, Mura G, Liberati A. Practice guidelines developed by specialty societies. The need for a critical appraisal. Lancet 2000; 355:103-6.

[17] Grimshaw JM, Russell IT. Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluations. Lancet 1993; 342: 1317-1322.

[18] Grimshaw J, Thomas R, MacLennan G, Fraser C, Ramsay CR, Vale L et al. Effectiveness and efficiencyof guideline dissemination and implementation strategies. *Health Technology Assessment* 2004; 6 (8): 1-72.

[19] Grol R, Dalhuijsen J, Thomas S, Veld C, Rutten G, Mokkink H. Attributes of clinical guidelines that influence use of guidelines in general practice: observational study. *BMJ* 1998;317:858-61

[20] Grol R, Grimshaw J. From best evidence to best practice: effective implementation of change in patients' care. *Lancet* 2003; 9391 (362): 1225-1230.

[21] Grol R. Improving the quality of medical care: building bridges among professional pride, payer profit, and patient satisfaction. *JAMA* 2001;286(20):2578-85.

[22] Grol R, Wensing M, Hulscher M, Eccles M. Theories on implementation of change in healthcare. in *Improving patient care: the implementation of change in clinical practice*. R. Grol, M. Wensing, M. Eccles, (eds). Elsevier, Edinburgh, 2005.

[23] Guyatt GH, Sackett DL, Sinclair JC, Hayward R, Cook DJ, Cook RJ, et al. Users' guide to the medical literature IX: a method for grading health care recommendations. *JAMA* 1995;274:1800-4.

[24] Hadorn DC, Baker D, Hodges JS, Hicks N. Rating the quality of evidence for clinical practice guidelines. *J Clin Epidemiol* 1996; 49: 749-754.

[25] Harbour R, Miller J. A new system for grading recommendations in evidence based guidelines. *BMJ* 2001;323: 334-6.

[26] Handley M, Stuart M, Kirz H. An evidence-based approach to evaluating and improving clinical practice. *HMO Pract* 1994; 4:75-83.

[27] Hunt DL, Haynes RB, Hanna SE, Smith K. Effects if Computer-based Clinical Decision Support Systems on Physician Performance and Patient Outcomes. A Systematic Review. *JAMA* 1998;280:1339-46.

[28] Lacasse Y, Ferreira I, Brooks D, Newman T, Goldstein RS. Critical appraisal of clinical practice guidelines targeting chronic obstructive pulmonary disease. *Arch Intern Med* 2001;161:69-74.

[29] Mainz J. Developing evidence-based clinical indicators: a state of the art methods primer. *Int J Qual Health Care* 2003; 15 suppl I: i5-i11.

[30] Mansfield C. Attitudes and behaviours towards clinical guidelines: the clinicians perspective. *Qual Health Care* 1995; 4:250-255.

[31] Marcos M, Berger G, Harmelen F van, Ten Teije A, Roomans H, Miksch S. Using critiquing for improving medical protocols: harder than it seems. In S. Qualini, P. Barahona, and S. Andreassen, editors, *Proc. of the 8th European Conf. on AI in Medicine (AIME-01)*, number 2101 in LNAI, pages 431–441, Germany, 2001. Springer Verlag.

[32] Marcos M, Roomans H, Ten Teije A, Harmelen F van. Improving medical protocols through formalization: a case study. In *Proc. of the 6th Int. Conf. on Integrated Design and Process Technology (IDPT-02)*, 2002.

[33] Mead P. Clinical guidelines: promoting clinical effectiveness or a professional minefield. *J Adv Nurs* 2000; 31:110-116.

[34] Mulrow CD. Rationale for systematic reviews. *BMJ* 1994; 309: 597-599.

[35] National Health Medical Research Council. A guide to the development, implementation and evaluation of clinical practice guidelines. Canberra: Commonwealth of Australia, 1998.

[36] Ollenschläger G, Marshall C, Qureshi S, Rosenbrand K, Burgers J, Mäkelä, Slutsky J. Board of Trustees 2002, Guidelines International Network (G-I-N). Improving the quality of health care: using international collaboration to inform guideline programmes by founding the Guidelines International Network (G-I-N). *Qual Saf Health Care* 2004;13(6):455-60.

[37] Rubin HR, Pronovost P, Diette GB. The advantages and disadvantages of process-based measures of health care quality. *Int J Qual Health Care* 2001; 13: 469–474.

[38] Shaneyfelt TM, Mayo-Smith MF, Rothwangl J (1999) Are guidelines following guidelines? The methodological quality of clinical practice guidelines in the peer-reviewed medical literature. *JAMA* 1999;281:1900-5.

[39] Shiffman RN, Dixon J, Brandt C, Essaihi A, Hsiao A, Michel G et al. The GuideLine Implementability Appraisal (GLIA): development of an instrument to identify obstacles to guideline implementation. *BMC Medical Informatics and Decision Making* 2005, 5:23doi:10.1186/1472-6947-5-23.

[40] US Department of Health and Human Services, Public Health Service, and Agency for Health Care Policy and Research. AHCPR Publication No. 95-0046, March 1995.

[41] Verkerk K, Van Veenendaal H, Severens JL, Hendriks EJ, Burgers JS. Considered judgement in evidence-based guideline development. *Int J Qual Health Care* 2006; 18(5):365-9.

[42] Wollersheim H, Hermens R, Hulscher M, Braspenning J, Ouwens M, Schouten J et al. Clinical indicators: development and applications. *Neth J Med.* 2007; 65(1):15-22.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-22

Chapter 2
Computer-interpretable Guideline Formalisms

Paul DE CLERCQ[a, 1], Katharina KAISER[b], Arie HASMAN[c]
[a]*Medecs, the Netherlands*
[b]*Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria*
[c]*Department of Medical Informatics, University of Amsterdam, the Netherlands*

Abstract. Implementing Computer-Interpretable Guidelines (CIGs) in active computer-based decision support systems promises to improve the acceptance and application of guidelines in daily practice. The model and underlying language are the core characteristics of every CIG approach. However, currently no standard model or language has been accepted by the CIG community. This aim of this chapter is to provide an overview of well-known approaches and to formulate a set of (minimal) requirements that can be used in the process of developing new CIG approaches or improving existing ones. It presents five CIG approaches (the Arden Syntax, GLIF, PRO*forma*, Asbru and EON), followed by a general discussion of the strong points of each approach as well as their implications for future research.

Keywords. Computer-interpretable Guidelines. Knowledge Representation. Decision Support Systems.

Introduction

Computer-interpretable Guidelines

During the last decade, studies have shown the benefits of using clinical guidelines in the practice of medicine such as a reduction of practice variability and patient care costs, while improving patient care. A variety of guidelines have been developed that focus on different application domains as well as different modes of use.

Although the potential application of guidelines in daily care is enormous, a number of difficulties exist related to the development and implementation of guidelines. One of them is the interpretation of the content of a guideline: the exact meaning of terms is not always defined, recommendations are not always clearly articulated and sometimes vague wording is used. Most of these guidelines are written down as large documents in a textual format, which are often cumbersome to read and difficult to integrate and apply in the patient care process. Additional problems exist also, related to the areas of maintenance (e.g., updating and versioning) and (local)

[1] Corresponding Author: Paul de Clercq, Medecs, Horsten 2, 5612 AX Eindhoven, The Netherlands; E-mail: p.a.d.clercq@medecs.nl

adaptation (e.g., adapting national guidelines to local protocols). Although the importance of guidelines is increasingly recognised, health care institutions often pay more attention to guideline development than to guideline implementation for routine use in daily care.

Implementing guidelines in active computer-based decision support systems promises to improve the acceptance and application of guidelines in daily practice because these systems are able to monitor the actions and observations of care providers and to provide guideline-based advice at the point of care. It is stated that (guideline-based) decision support systems are in fact necessary for the future of medical decision making in general [1].

These so-called Computer-Interpretable Guidelines (CIGs) are increasingly applied in diverse areas and many parties are developing CIGs as well as decision support systems that incorporate these guidelines, covering a wide range of clinical settings and tasks (an overview can be found on OpenClinical [2]). Despite these efforts, only a few systems progressed beyond the prototype stage and the research laboratory. Building systems that are both effective in supporting clinicians and accepted by them has proven to be a difficult task.

Various questions arise when developing and implementing CIGs, such as:

- How to represent and share various types of guidelines using a formal and unambiguous representation;
- How to acquire, verify, localize, execute and evaluate formalised guidelines and support systems in daily practice;
- How to interface guideline-based decision support systems with external patient information systems;
- How to provide decision support to a care provider in daily practice.

Although these are all relevant and important questions, this chapter will focus mostly on the first questions, namely with respect to the issue of representing and sharing CIGs using a formal model. More information on other issues concerning the development and implementation of CIG decision support systems is described elsewhere [3].

CIG Approaches

Nowadays, many approaches exist for specifying CIGs, each with its own motivations and features [4]. For example, some approaches focus more on guideline standardisation and interoperability, while others focus more on guideline development or decision support. These different foci have their implications for the representation of CIGs.

This chapter will present and discuss a number of well-known CIG approaches, with the goal of providing a general comparison and discussion in order to identify the strong points of the various CIG approaches. Based on known approaches, reviews [3, 4, 5-8] and own experiences, it is possible to define the functionality of CIG approaches in terms of a two main characteristics: the underlying model and the language in which guidelines are specified.

The model is the core characteristic of every guideline approach. It must be able to represent various kinds of guidelines that may differ considerably in complexity and level of abstraction, for example by means of nesting or decomposition. The model must contain a set of building blocks used to construct guidelines, such as tasks, rules, nodes or frames. For example, most approaches model guidelines in terms of a Task-Network Model (TNM): a (hierarchical) model of the guideline control flow as a network of specific tasks (e.g., flowchart) [6]. TNMs are typically based on a standard repertoire of generic tasks such as decisions and actions. The model must be expressive enough to represent these various aspects. Also, guidelines contain a number of different knowledge types such as declarative knowledge (e.g., domain-specific knowledge) and procedural knowledge (e.g., inference or the method of decision support). The model must support these types of knowledge and should model them separately to support guideline sharing and to ensure that guidelines can be used in multiple clinical domains and in various modes (e.g., proactive vs. reactive use) [9]. The model should also support aspects related to didactics and maintenance: as the content of a guideline is not static but may change over time, the representation must be able to store didactic and maintenance information such as author names, versioning information, purposes and detailed explanations.

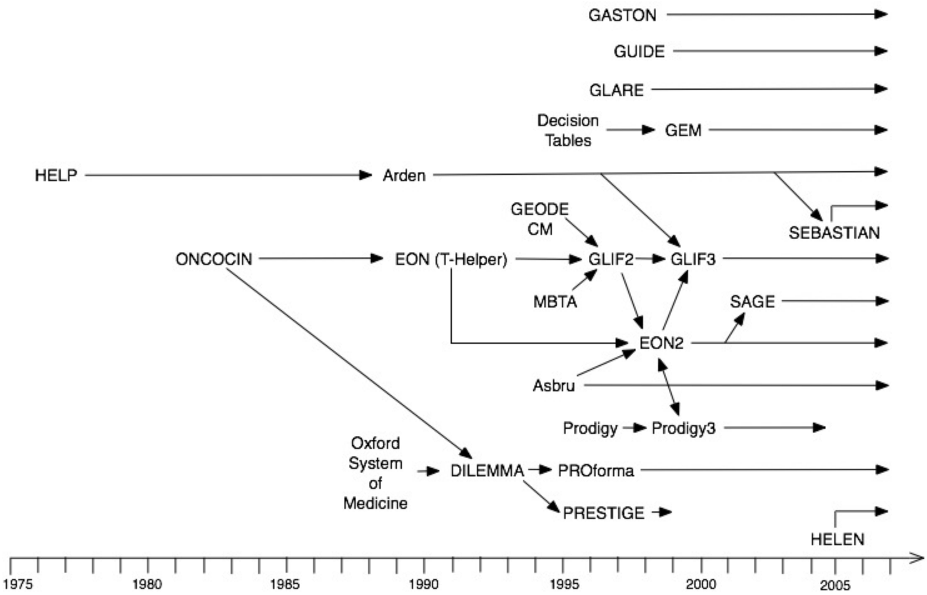


Figure 1. History of CIG approaches, positioned on a time axis (adapted from Elkin et al [10])

The guideline model should be supported by a formal language (vocabulary, syntax and semantics) which specifies the actual guidelines in terms of the above-mentioned model constructs. Usually, such a language consists of two parts: a control-flow language and an expression language. The control-flow language usually specifies the guideline structure (flow) in terms of constructs of the model (e.g., the various tasks of the above-mentioned TNM), whereas the expression language usually describes decision criteria (e.g., ‘is the patient older than 65 years’). This formal language (including both parts) must be interpretable by automatic parsers. Preferably, each approach should include a guideline execution engine, which incorporates such a parser that is able to provide decision support based on the encoded guidelines.

This chapter will compare and discuss a number of approaches in terms of these characteristics. The approaches discussed in this chapter are selected on the basis of information from OpenClinical [2] and the knowledge of the authors on existing approaches. Inclusion of an approach was based on the following criteria. First of all, as this chapter aims at identifying and discussing CIG approaches in terms of different characteristics (e.g., model, language), we selected approaches that differ as much as possible with respect to these characteristics. Furthermore, we used criteria such as lifetime of the approach and number of publications and whether the approach is considered generally as a 'key' approach. The final inclusion of an approach as a relevant subject was based on a subjective decision. Therefore, although we recognize that a number of other important approaches exist nowadays [2] such as PRODIGY, GUIDE, Gaston, GLARE, SAGE, HELEN, DeGel and SEBASTIAN, we have limited the number of refereed approaches (also to constrain the chapter's size) to the following five: The Arden Syntax [11], GLIF [12], PRO*forma*, [13], Asbru [14] and EON [15].

The remaining part of this chapter describes each of the five approaches, after which all approaches are compared in terms of the above-mentioned characteristics. The chapter finishes with a general discussion on guideline approaches, their strong points and their implications for future research.

1. The Arden Syntax

1.1. Introduction

Named after the Arden Homestead Conference Centre, where the initial meeting was held, the first version of the Arden Syntax was developed in 1989 [11] as a response to the inability to share medical knowledge among different institutions. The Arden Syntax is intended as an open standard for the procedural representation and sharing of medical knowledge. It defines a representation for modular guidelines: Medical Logic Modules (MLMs) [16]. The Arden Syntax focuses on the sharing of 'simple' modular and independent guidelines (e.g., reminders). It is not designed for complex guidelines that for example address treatment protocols [17]. The Arden Syntax was accepted in 1992 as a standard by the American Society for Testing and Materials (ASTM). The current version of the Arden Syntax is Arden 2.0 [18], developed and published by the HL7 group. The Arden Syntax has been used by different institutions and companies to develop and implement guidelines in multiple clinical settings.

1.2. Model

1.2.1. Medical Logic Modules

In the Arden Syntax, guidelines are modelled as (a collection of) Medical Logic Modules (MLMs). Each MLM represents a single decision and contains slots that are grouped into three categories: *Maintenance*, *Library* and *Knowledge*. The *Maintenance* and *Library* categories describe the MLM's pragmatics (e.g., title, version, explanation and keywords) and the *Knowledge* category describes the logic of an MLM. Figure 2 shows an example of (part of) an MLM that warns a health care provider whenever a patient's hematocrit value becomes too low. The remaining part of this section will explain the various parts of an MLM in more detail.

```
maintenance:
  title: Alert on low hematocrit;;
library:
  purpose: Warn provider of new or worsening anemia.;;
knowledge:
  type: data-driven;;
  data:
    blood_count_storage := event {'complete blood count'};
    hematocrit := read last {'hematocrit'};
    previous_hct := read last ({'hematocrit'} where it occurred before
                    the time of hematocrit);;
  evoke: blood_count_storage;;
  logic:
    if hematocrit is not number then conclude false; endif;
    if hematocrit <= previous_hct-5 or hematocrit<30 then conclude true;
    endif;;
  action:
    write " The patient's hematocrit ("|| hematocrit ||") is low or
            falling rapidly.";;
end:
```

Figure 2. An example of an MLM

1.2.2. Maintenance and Library Slots

As MLMs are to be shared among various institutions, the *Maintenance* and *Library* categories contain necessary documentation for each MLM. The *Maintenance* slots include the MLM's (file)name, author, version, institution, specialist, date of last modification and validation status (e.g., '*testing*', '*research*', '*production*' or '*expired*').

The slots in the *Library* category are used for documentation and consist of the MLM's purpose, a more detailed explanation (which can for example be shown to users when they receive MLM-generated messages) and a number of keywords (for example used to categorize MLMs).

1.2.3. Knowledge Slots

The actual medical knowledge is stored into the *Knowledge* category. This category consists of five mandatory slots (*type*, *data*, *evoke*, *logic* and *action*) and two optional slots (*priority* and *urgency*). Of these slots, the most important ones are *data*, *evoke*, *logic* and *action*.

The *data* slot is used to obtain the values of concepts that are mentioned in the MLM from local clinical information systems such as Electronic Patient Record (EPR) systems. For example, the line '*hematocrit := read last {'hematocrit'};*' indicates that the value of the concept '*Hematocrit*' (used in the logical expression of the MLM in figure 2) corresponds to the last hematocrit value in for example an EPR. The terms between the curly braces are often institution-specific: the implementation and integration of the actual interface techniques are usually left to the local institutions [19].

The *evoke* slot specifies the context in which an MLM should be executed. MLMs can be executed as a result of three different types of events: database operations, temporal events and external notifications. The first one is most commonly used. For example, the MLM in figure 2 is executed as a result of the '*blood_count_storage*' event (i.e., whenever a new blood count is added to the system's database).

The *logic* slot contains the actual decision criteria that may lead to a certain action. These logical expressions are implemented as production rules and contain concepts that are defined in the *data* slot (e.g., '*Hematocrit*'). The Arden Syntax supports various types of operators such as logical operators, list operators, temporal operators and aggregation operators. The Boolean operators use a three-valued logic, in which the value '*null*' is considered as unknown. Whenever the rule's premise is evaluated '*true*', a particular action that is specified in the *action* slot is carried out. When the premise is evaluated '*false*' or '*null*', the execution of the MLM ends.

Once the logical expression evaluates to '*true*', the *action* slot is executed, performing whatever actions are specified in this slot. Typical actions include sending a message to a health care provider, adding an interpretation to the patient record, returning a result to a calling MLM, and evoking other MLMs (nesting). For example, the MLM in figure 2 writes a message to the standard destination, stating that the patient's hematocrit value is low or falling (the || operator is a concatenation operator, inserting the actual hematocrit value of the patient into the message).

The TNM of a single MLM always consists of three steps: the *evoke* slot determines whether the *logic* slot should be executed, which on its turn determines whether the *action* slot should be carried out. Although it is possible for an MLM to invoke other MLMs by means of the '*call*' statement in the *action* slot, the approach does not support a formal TNM to steer these invocations [20].

1.3. Language

The Arden syntax TNM is formally defined in Backus-Naur Form (BNF). MLMs are text-based (each MLM is encoded as an ASCII file) and always have the format, shown in figure 2. The expression language encodes criteria (in the *logic* slot) as textual production rules.

The approach does not contain a standard execution engine that is able to interpret and execute guidelines. However, a number of implementations for executing MLMs have been developed, including the use of pseudocode [21], C++ [22] and MUMPS [23]. As the Arden Syntax leaves the implementation of patient data modelling entirely up to the local institutions, there are no standard mapping facilities to obtain values of required patient data during guideline execution.

2. The GuideLine Interchange Format (GLIF)

2.1. Introduction

The GuideLine Interchange Format (GLIF) was developed to model guidelines in terms of a flowchart that consists of structured scheduling steps, representing clinical actions and decisions. Figure 3 shows an example of a GLIF guideline (aimed at the treatment of chronic cough), visualised through the Protégé knowledge modelling tool [24].

GLIF was developed by the Intermed Collaboratory [25] including researchers at Columbia University, Harvard University and Stanford University and was first published in 1998 [12]. The intended purpose of GLIF is to facilitate sharing of guidelines between various institutions by modelling guidelines in such a manner that the guidelines are understandable by human experts as well as by automatic parsers used in different clinical decision support systems. The current version of GLIF is GLIF3 [26], which is discussed in the remaining part of this section.

A variety of guidelines [8, 27, 28] have been specified using GLIF to evaluate the various aspects of the approach.

2.2. Model

2.2.1. Multi-level Approach

In order for guidelines to be 1) readable by humans, 2) interpretable by computers and 3) adaptable by different (local) institutions, GLIF defines a specification of a guideline at three levels of abstraction: the conceptual level, the computable level and the implementable level.

The highest level is the conceptual level where guidelines are represented as flowcharts, which can be viewed by humans (e.g., guideline authors) but are not interpretable by decision support systems. At this level, details such as the contents of patient data elements, clinical actions and guideline flow are not formally specified.

These specifications are provided at the computable level. At this level, the guideline content is formally defined and various verification checks of the guidelines are carried out (this level is described in more detail in section 2.2.2). Finally, at the implementable level, guidelines can be custom-tailored to particular institutional information systems. At this stage, institution-specific procedures and mappings (which are usually non-sharable) are specified (see also section 2.2.3).

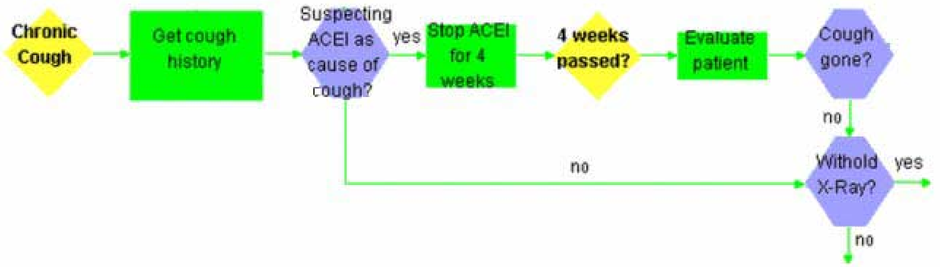


Figure 3. Graphical representation of a treatment chronic cough treatment guideline in GLIF (adapted from Boxwala et al [26])

2.2.2. The Computable Level

In contrast to the conceptual level, where guidelines are visualised merely as graphical flowcharts, the computable level allows for a formal specification, using the GLIF TNM. This model is object-oriented and consists of a number of classes that describe typical guideline tasks (e.g., decisions and actions).

The *Guideline* class represents a (sub)guideline. Each guideline is modelled as an instance of this class. The *Guideline* class contains a number of attributes that are administrative in nature (e.g., name and author) but also attributes that describe the capabilities of a guideline (e.g., the guideline's intention). In addition, it also contains a reference to a collection of steps that are linked together in a directed graph (flowchart). Similar to a guideline, steps are also represented by classes, and each step in a guideline is an instance of such a class. GLIF defines five classes that represent the following steps: *Decision* steps, *Patient state* steps, *Branch* steps, *Synchronization* steps and *Action* steps.

Decision steps model decision points in a guideline and direct flow control from one guideline step to various alternatives. There are two types of *Decision* steps: *Case* steps and *Choice* steps. A *Case* step is a *Decision* step that contains a number of logical expressions and thus is used to model deterministic decisions. Based on the outcome, the guideline flow is directed to the various alternatives. In contrast, *Choice* steps represent situations where a guideline suggests preferences, but leaves the actual choice to an external agent. *Choice* steps contain rules that support or oppose the various preferences.

A *Patient state* step serves as a label that describes the current patient state that results after having carried out previous steps. It can also be used as an entry point in the guideline, depending on the current patient's state (e.g., the patient revisits a family practitioner with a high blood pressure). Each *Patient state* step contains attributes that describe the state of the patient (e.g., the blood pressure is higher than 140/90 during the last week). Whenever this state occurs in practice, the guideline that contains the corresponding *Patient state* step is executed.

Branch steps model a set of concurrent steps by directing flow to multiple parallel guideline steps and are used in conjunction with *Synchronization* steps. Multiple guideline steps that follow a *Branch* step always eventually converge in a corresponding *Synchronization* step. When a certain branch reaches the corresponding *Synchronization* step, a *continuation* attribute specifies whether all, some, or one of the preceding steps must have been completed before control can move to the next step.

Action steps model actions that have to (or should) be performed. Three types of actions are defined: 1) medically oriented actions such as a recommendation for a particular course of treatment, 2) programming-oriented actions such as retrieving data from an electronic patient record or supplying a message to a care provider, and 3) control-oriented actions that invoke nested structures such as (sub)guidelines or macros to support recursive specification. For example, GLIF defines an MLM-macro, which can be used to define an MLM. Internally, the macro consists of two steps: a *Decision* step and an *Action* step.

2.2.3. The Implementable Level

Similar to the Arden Syntax, decision criteria and action specifications in GLIF contain references to actual patient data (e.g., the age of a patient) and medical concepts (e.g., antibiotic, amoxicillin), which have to be acquired during guideline execution from patient information systems. In order to facilitate sharing of guidelines among different institutions, this information is stored in the implementable level. This level contains the information to integrate developed guidelines with institution-specific medical knowledge sources and information systems such as EPRs. GLIF aims at defining the structure of patient data elements and medical concepts in this level in accordance with standard data models and medical terminologies such as HL7's Reference Information Model (RIM) [29] or the Unified Medical Language System (UMLS) [30]. The implementable layer is currently being further developed [31].

2.3. Language

In GLIF, the guideline TNM itself (e.g., all classes, attributes and relations) is described by means of Unified Modeling Language (UML) class diagrams. The control-flow language that describes the actual guidelines (in terms of class instances) is the Resource Description Format (RDF) language.

Decision criteria are specified through a formal expression language, referred to as the Guideline Expression Language (GEL) [32], which is a superset of the Arden Syntax. In addition, the object-oriented expression language GELLO has been developed to specify decision criteria in GLIF [33]. In contrast with the original GEL language, the GELLO language is able to include references to concepts and attributes from the core GLIF model. The GELLO standard has recently been accepted as a standard expression language by HL7 and ANSI.

A guideline execution engine named GLEE (GuideLine Execution Engine) has been developed which is able to execute GLIF-encoded guidelines and can be integrated into the clinical information system of a local institution [34].

A separate approach is the development of a generic guideline execution engine, named the Guideline Execution by Semantic Decomposition of Representation (GESDOR) [35], which is able to execute various control-flow and expression languages, GLIF being one of them. The GLIF model and language are still being further developed.

3. PRO*forma*

3.1. Introduction

PRO*forma* is a CIG approach supported by acquisition and execution tools with the goal of supporting guideline dissemination in the form of decision support systems that assist patient care through active decision support and workflow management [13]. PRO*forma* was initially developed at the Cancer Research UK Advanced Computation Laboratory. The name PRO*forma* is a concatenation of the terms proxy (‘authorised to act for another’) and formalize (‘give definite form to’).

Similar to GLIF, PRO*forma* also represents guidelines as a directed graph in which the nodes are instances of a fixed set of classes. Figure 4 shows an example of a guideline in terms of instances of these classes, visualised through the Arezzo Composer, part of the Arezzo suite, developed by Infermed Ltd. [36].

Besides the commercially available Arezzo suite, a second suite has been developed to acquire and implement PRO*forma* guidelines, called the Tallis suite [37].

A large amount of CIG and CIG-based decision support systems have been developed in various areas by means of the Arezzo and Tallis suites [38].

3.2. Model

The PRO*forma* TNM is called the PRO*forma* task ontology. Each guideline in PRO*forma* is modelled as a plan that consists of a sequence of tasks. The PRO*forma* task ontology defines four classes, each with their own attributes: *Plans*, *Decisions*, *Actions* and *Enquiries*. These four tasks are derived from the generic *Keystone* task, which contains a number of attributes that are common to all four derived tasks. These include administrative ones that hold a name, caption, or description but also attributes that describe the capabilities of a task such as goals and conditions.

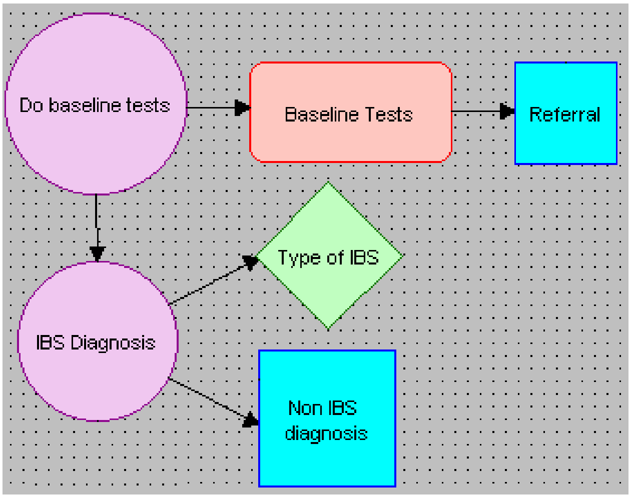


Figure 4. Part of a PRO*forma* guideline

Each *Plan* models a (sub)guideline. *Plans* define 1) an ordered sequence of tasks, 2) logical and temporal constraints on their enactment and 3) circumstances in which a plan must be aborted or terminated (e.g., exceptions). Besides the common attributes that are defined in the *Keystone* task, the plan task contains additional attributes that store the plan's task network, scheduling and temporal constraints and abort or termination conditions.

The plan's network is stored as a set of task instances (similar to the *Guideline* class in GLIF). For example, a guideline that consists of four task instances (e.g., '*history*', '*diagnosis*', '*therapy*', '*follow-up*') is modelled through a *Plan* instance that contains references to those four task instances.

The ordering between these task instances is defined by means of two sorts of constraints: scheduling constraints and temporal constraints. Scheduling constraints order tasks in a plan by means of qualitative conditions (e.g., the '*history*' task is executed '*before*' the '*diagnosis*' task). Temporal constraints order tasks by using temporal conditions (e.g., the '*follow-up*' task is executed '*after a period of ten weeks*'). By using these two types of constraints, tasks in a plan are modelled differently than traditional flowcharts that order guideline elements usually only through scheduling constraints.

Another way of directing guideline flow in PRO*forma* is through abort or termination conditions. Each PRO*forma* task passes through a number of states such as '*dormant*', '*in progress*', '*aborted*', '*terminated*' and '*performed*'. Every task is initially in a '*dormant*' state. Executing a certain task changes its state from '*dormant*' to '*in progress*'. Whenever a task is finished normally, the task's state becomes '*performed*'. It is possible to force the termination or abortion of a plan by means of the abort and termination conditions.

A *Decision* task is represented as a set of possible outcome candidates plus various types of schemas (logical expressions) that support or oppose each candidate. Every candidate is associated with a set of schemas. Schemas consist of rules, qualitative variables, quantitative weightings and certainty factors [39] and support (+) or oppose (-) candidates, establishing a preference order among the candidates.

An *Action* is a task that a PRO*forma* execution engine can request for enactment by an external agent (e.g., a clinical user or an external software program or hardware device). Such an action in PRO*forma* usually exists of issuing a message to a user or calling an external program through a predefined Application Programming Interface (API). Examples are '*"give ibuprofen, 10 mg"*' that shows a message to a clinical user or '*call(print(leaflet1))*' that executes an external procedure to print a leaflet. In PRO*forma*, actions are always atomic and are not decomposable.

Enquiries are used to acquire various kinds of information, such as clinical or administrative information. This information can be obtained from a clinical user or can be directly extracted from an external software agent or hardware device (e.g., EPR or patient monitor). Therefore, as was the case with the definition of an action, the *Enquiry* class contains attributes that define the method of data retrieval.

3.3. Language

Guidelines in PRO*forma* are stored (as instances of the PRO*forma* task ontology) using a language, derived from the so-called Red Representation Language (R^2L), a time-oriented control-flow language [40]. In PRO*forma*, a guideline is a declarative specification of tasks and their (inter)relationships organised in a hierarchy of plans and their components. This language also contains a formal expression language to express goals (e.g., '*achieve(normal_respiration)*'), conditions (e.g., '*peak_flow < 30*', '*risk_level = severe*') and argument schemes ('*diagnosis = oesophagitis and liver_disease = absent then cimetidine: +*').

Before execution, guidelines are translated into another language, called L_{R2L} ('Logic of R^2L'), a language based on predicate logic. This language is used as input for the execution module.

Execution of PRO*forma* guidelines is supported by means of two execution engines, which are part of the earlier mentioned Arezzo and Tallis suites. In addition, the earlier-mentioned generic execution engine GESDOR [35] is also able to execute PRO*forma* guidelines.

4. Asbru

4.1. Introduction

Asbru is a CIG approach, developed at Stanford University, the Vienna University of Technology and the Ben-Gurion University, which focuses on the application and critiquing of time-oriented clinical guidelines [14]. This approach aims at representing clinical guidelines as time-oriented skeletal plans, which are plan schemata at various levels of detail. In order to manage these (often complex) skeletal plans, key aspects of Asbru are the representation of high-level goals (intentions), the representation of temporal patterns and time annotations, and the development of user interfaces to visualize developed plans. An example of an Asbru guideline can be shown in figure 4 of Chapter 8.

4.2. Model

The Asbru TNM represents guidelines as skeletal plans. Similar to the notion of plans in PRO*forma*, a plan is a collection of other (sub)plans and/or actions. The Asbru TNM consists of a number of elements, of which the *Plan* element is the most important one. Besides administrative attributes (e.g., the plan's title), each plan contains the following attributes (referred to as knowledge roles in Asbru), which describe each plan's functionality: *preferences*, *intentions*, *conditions*, *effects* and *plan body*.

Preferences bias or constrain the applicability of a plan to achieve a certain goal. Examples of *Preferences* are 1) '*select-method*', a matching heuristic to determine the applicability of the entire plan (e.g., '*exact-fit*' or '*roughly-fit*'), 2) '*resources*', a specification of forbidden or obligatory resources (e.g., in certain cases of a pulmonary infection treatment, surgery is prohibited and antibiotics must be used), and 3) the applied '*strategy*' (e.g., '*aggressive*' or '*normal*').

Intentions are used to model the aims of the plan, independent of the plan body. Intentions can for example aid in the selection of the most appropriate plan by for example defining which patient state(s) must hold during of after a plan's execution (e.g., the patient's blood pressure must never exceed 140/90) or which actions should take place (e.g., maintain monitoring of blood glucose once a day). Intentions are modelled as temporal patterns.

Conditions are also temporal patterns and are used to change the state of a plan. In Asbru, similar to the PRO*forma* approach, plans are in a certain state during execution time (e.g., '*activated*', '*suspended*', '*aborted*' and '*completed*'). Asbru defines a number of condition categories such as '*filter-preconditions*' and '*setup-preconditions*' that need to hold if a plan is considered applicable, '*suspend-conditions*' that determine when an active plan must be (temporarily) suspended, '*abort-conditions*' that determine when an active or suspended plan has to be aborted and '*completed-conditions*' that determine when a plan is (successfully or not) completed.

Besides the earlier-mentioned *intentions* and *conditions*, *effects* can also be used to select the most appropriate plan by describing the expected behavior of the plan's execution. For example, a treatment plan might decrease the blood-glucose level, which classifies this plan as not the most appropriate for a certain class of patients. Effects may include probabilities that specify the probability of the effect's occurrence.

The *plan body* is a set of subplans or actions (which are plans that do not contain any subplans anymore) that have to be performed whenever the plan is considered appropriate (based on the plan's preconditions, intentions or effects). The order in which the subplans are executed is determined by the *Plan*'s *type* and *subtype* attributes. Asbru defines four types for synchronising subplans: '*sequentially*', '*parallel*', '*any-order*', and '*unordered*', which are described by means of the subplans' *type* attribute. Furthermore, cyclical plans can be defined. As mentioned earlier, plans that have been started can be suspended, aborted or completed (based on the plan's conditions).

4.3. Language

The Asbru TNM itself is defined as a Document Type Definition (DTD), which defines the structure of the various elements of the TNM. The control-flow language and the expression language are formally defined by means of XML, based on the earlier-mentioned DTD [41].

An important aspect of the expression language is the concept of time annotations, which are used in specifying complex temporal patterns, and specify the temporal constraints within which an action must take place, or a condition must be fulfilled in order to trigger. The Asbru expression language also supports the concept of temporal abstractions such as 'has the patient suffered from a second episode of anaemia of at least moderate severity'.

Beside the above-mentioned XML language, other languages, related to Asbru such as MHB (Many-Headed Bridge) are being developed with the goal of bridging the gap between informal (e.g., textual) and formal guidelines [42].

Various execution engines have been developed that are able to execute (a subset of) Asbru guidelines such as the Asbru interpreter [43], developed as part of the Protocure-II project [44] and the Asbru Execution Engine [45]. Another execution engine that has been developed to execute (simplified) Asbru guidelines is the Spock system [46], which is part of the DeGel framework [47].

5. EON

5.1. Introduction

EON, developed at Stanford University, is a CIG approach that aims at developing decision support systems that reason about guideline-directed care [15]. The EON approach consists of several components that facilitate the acquisition and execution of clinical guidelines.

Similar to GLIF, the EON TNM, called Dharma [48], is object-oriented and consists of classes that describe guideline tasks as a sequence of structured temporal steps. The Dharma model is non-monolithic, meaning that it can be extended with additional classes that capture new guideline behaviour. Besides the Dharma guideline model, the EON architecture also contains a number of run-time components, used to construct execution-time systems. An example of a guideline in EON, visualised (similar to GLIF guidelines) through the Protégé knowledge modelling tool [24] can be seen in figure 1 of chapter 8.

The EON project has been discontinued since 2002, but was succeeded by the SAGE project, which ran from 2002-2006 [49]. One of the CIG systems that were developed using EON is the ATHENA system, which addresses the treatment of hypertension, and is still in clinical use today [50].

5.2. Model

5.2.1. The Dharma Guideline Model

In contrast with for example GLIF and PRO*forma* that model guidelines in terms of a fixed number of classes (e.g., decisions, actions), the researchers of EON propose a non-monolithic (non-fixed) TNM, which consists of a standard set of classes that can be extended with task-specific submodels, resulting in additional classes that are matched to the knowledge requirements of different guidelines.

The EON approach aims at modelling multi-encounter patient management (e.g., chronic disease management), in which each guideline represents a certain state of the patient and consists of a number decisions and actions that are applicable to that patient state and may lead to changes in patient states over time (and may also trigger other guidelines as a result). In order to define guidelines according to this conceptual model, the TNM consists of a number of standard classes and attributes which form the so-called core guideline ontology. The four most important classes are *Scenarios*, *Decisions*, *Actions* and *Activities*.

A *Scenario* is a (partial) characterisation of the state of a patient (e.g., the patient is currently being prescribed a low-dosed steroid). In a scenario, eligibility conditions specify the necessary conditions for a patient to be in this scenario. Scenarios –which were firstly introduced as part of the PRODIGY guideline approach [51]– allow a clinician to synchronize the management of a patient with the corresponding parts of (a portion of) a guideline and are commonly used as entry points in a guideline. In the Dharma ontology, a scenario is always followed by a decision or action step. Each scenario in an actual guideline is an instance of the *Scenario* class, which contains several attributes such as an attribute that specifies the eligibility criteria and an attribute that specifies the step that follows the current scenario (similar to GLIF). Scenarios allow a clinician to synchronize the management of a patient to situations handled by a guideline and can also serve to model exceptions, which represent exceptional situations that rarely occur. As expressing everything in a guideline can be impractical, a guideline author may want to partition the guideline into normal situations that cover usual cases and exceptions.

In the Dharma core ontology, two basic types of *Decisions* are defined (by means of two subclasses): decisions that model '*if-then-else*' choices and decisions that require making a heuristic choice from a set of pre-enumerated alternatives. The latter is aided by preferences as determined by rule-in and rule-out conditions that support or oppose alternatives (similar to the concept of schemas in PRO*forma*).

Actions are instantaneous acts that lead to changes in the state of the world such as collecting patient data, displaying a message to the user or starting a drug regimen. Actions are used heavily throughout guidelines modelled in EON. Whereas actions refer to instantaneous acts, activities model processes that take place over time. *Activities* have states that can change from time to time. These changes are usually the result of actions specified in a guideline, as actions are able to start a new activity, stop an ongoing activity or change the attribute values of an ongoing activity. Finally, the model also includes actions that refer to a set of other actions or a subguideline. Similar to GLIF, examples of such actions are actions that model branching and synchronisation constructs in order to execute parallel tasks.

Every class in the Dharma ontology can be associated with a goal. The notion of goals is comparable with the notion of intentions in Asbru, although less sophisticated. In the Dharma ontology, goals are represented as Boolean criteria (e.g., '*reduce the arterial bloodpressure to less than 130/85 within three weeks*').

5.2.2. The Patient Data and Medical-specialty Model

The patient data model defines classes and attributes in order to represent patient data. It defines characteristics regarding demographic and clinical conditions of specific patients. It does not aim at modelling the entire patient (e.g., replicate the structure of an EPR), but models only those distinctions that are relevant for the purpose of defining guidelines and protocols.

The medical-specialty model consists of a medical domain ontology that models the structure of domain concepts (e.g., drugs and treatments) in terms of organised classes, relations and attributes. The medical-specialty model represents different sorts of domain-specific information.

5.3. Language

The Dharma TNM as well as the control-flow language is described by means of the internal frame-based Resource Description Format (RDF) of Protégé. Although the focus of the EON approach is not on defining a formal control-flow language, it does particularly address the subject of defining criteria in formal expression languages. EON defines three different expression languages.

First, common but relatively simple criteria can be expressed as Boolean criteria in terms of a set of object templates such as '*diabetes mellitus is present and the most recent serum creatinine is less than normal*'.

According to the researchers of the EON approach, such a criterion language is not expressive enough to capture more complex criteria such as '*is an authorised medication present that is contraindicated by some medical condition*'. To represent such criteria, the Protégé Axiom Language (PAL) is used, which is embedded into the Protégé development environment.

Finally, it is possible to write complex temporal criteria such as '*presence of an episode of uncontrolled blood pressure that overlaps with lisinopril medication and that started within two weeks after the initiation of lisinopril*'. These are written as temporal queries, which during guideline execution are translated to database queries. To specify temporal aspects, EON has adopted a subset of the Asbru temporal expression language to represent temporal information.

To facilitate the development of guideline execution engines, EON defines an extensive execution architecture that contains a guideline execution engine plus components for interfacing to third-party information systems [52].

6. Discussion

6.1. Comparison

6.1.1. Overview

Each approach focuses on different aspects of guideline modelling and representation, which have their implications regarding the representation and modelling of guidelines as shown in the previous sections. This section will address the strong points of each approach, after which a number of (minimal) requirements are formulated that were distilled from these points that can be used in the process of developing new approaches or improving existing ones.

6.1.2. Model

All approaches described in this chapter use a TNM that models guidelines in terms of a sequence of class instances (e.g., flowchart), with the exception of the Arden Syntax that models guidelines as (a collection of) independent modular rules. As a result, the Arden Syntax is most suitable for representing simple guidelines such as alerts in reminder systems, but less suitable for complex multistep guidelines.

Although the terminology may differ, all approaches support a basic set of ‘core’ guideline tasks, such as decisions, actions and entry criteria. Decisions for example are represented by means of *logic* slots in the Arden Syntax, *Decision* steps in GLIF, *Decision* tasks in PRO*forma*, *conditions* in Asbru, and *Decisions* in EON. Similarly, actions are represented by means of *action* slots in the Arden Syntax, *Action* steps in GLIF, *Action* and *Enquiry* tasks in PRO*forma*, *actions* (atomic *Plans*) in Asbru, and *Actions* in EON. Entry criteria are represented by *evoke* slots in the Arden Syntax, *Patient state* steps in GLIF, *preferences*, *intentions* and *effects* in Asbru, *triggers* and *conditions* in PRO*forma* [53] and *Scenarios* in EON.

The TNMs of all approaches define a fixed set of guideline tasks, with the exception of EON that is extensible.

All approaches described in this paper except for the Arden Syntax provide explicit support for controlled nesting of guidelines in order to model complex guidelines in terms of subguidelines (GLIF and EON) or subplans (PRO*forma* and Asbru). For this purpose, GLIF, EON and PRO*forma* contain an *Action* task that may contain a reference to a subguideline or subplan. In Asbru, each plan body contains a number of subplans until a non-decomposable plan (also called *Action*) is encountered. Although the Arden Syntax supports a form of nesting by calling other rules in the *Action* slot, there is no general control flow that controls these invocations. All approaches support the concept of referenced subguidelines. GLIF also supports the representation of common guideline structures through Macros, which facilitates the reuse of guidelines that are used often (e.g., ‘*if-then*’ rules such as MLMs).

EON, PRO*forma* and Asbru also support the use of goals and intentions to formally specify a guideline on a higher level of abstraction. Of these techniques, the Asbru intention model is the most sophisticated. GLIF defines different layers of abstraction, which allows guideline authors to view only the general control flow (flowchart) of a guideline before specifying all the necessary details. EON uses a non-monolithic approach: the Dharma guideline model is based on a core model, which can be extended with submodels depending on the complexity of the guideline.

Besides the knowledge that defines the guideline control flow (in terms of for example, rules, steps, plans), every guideline also contains domain-specific knowledge such as medical knowledge (e.g., terminology) and knowledge concerning the patient (e.g., the patient’s symptoms or history).

In the Arden Syntax each reference to a domain-specific item is stored as a label in the *data* slot of an MLM. As a result, an MLM does not ‘know’ for example that amoxicillin is an antibiotic. Also PRO*forma* and Asbru contain no explicit support for modelling domain-specific knowledge or for using standard terminology systems. GLIF addresses this problem by modelling domain-specific knowledge through the implementable level and EON takes a similar approach by defining, the Patient Data and the Medical-Specialty models.

Besides invoking subguidelines, a guideline may consist of various types of actions such as medically oriented actions (e.g., recommending a particular course of treatment) and programming-oriented actions (e.g., supplying a message to a care provider). In the Arden Syntax, actions (stored in the *action* slot) are usually programming-oriented as they are used to generate reminders or alerts. This is also the case in the PRO*forma* approach, as a PRO*forma* action is a programming-related task that is carried out by the execution engine through an Application Programming Interface (API). GLIF and EON both support these two types of actions. Finally, Asbru does not support programming-related actions.

Didactic and maintenance information concerns information about authors, versioning, purposes and detailed explanations. The Arden Syntax, GLIF and EON approaches are all able to hold various kinds of information such as the guideline's author, version, institution, keywords, validation (e.g., '*research*', '*testing*', '*production*') and explanation. In PRO*forma* and Asbru, it is not possible to store didactic- and maintenance-related information (besides a name and explanation).

6.1.3. Language

All approaches define a control-flow language that describes the guideline control flow (in terms of TNM constructs). All approaches except EON have also defined the TNM in a formal way using BNF, XML or UML. For each approach, the control-flow language supports all constructs of the corresponding TNM. The Arden Syntax describes guidelines using formatted text, GLIF and EON using RDF, Asbru using XML and PRO*forma* using R^2L. PRO*forma* is the only approach, which makes a distinction between a declarative language (R^2L), used during the guideline definition phase and a procedural language (L_{R2L}) that is processed during the guideline execution phase. In order to facilitate this translation, the PRO*forma* representation language contains constructs that are filled in during guideline acquisition but are execution-related. For example, PRO*forma* defines an execution state that denotes the state of a guideline during execution (e.g., '*in progress*', '*aborted*', '*terminated*', '*performed*'). This is in contrast with EON and GLIF that define patient states which are used during execution to determine the applicability of a guideline (as mentioned earlier, PRO*forma* is also able to model patient states implicitly through constructs like triggers and conditions). Similar to PRO*forma*, Asbru also uses the concept of guideline execution states.

All approaches also define formal expression languages that describe decisions and entry criteria. An important aspect of these languages is the issue of temporal reasoning. All approaches support some form of temporal reasoning, of which the Asbru approach contains the most sophisticated structures. EON and GLIF both adopt a subset of the Asbru temporal language. In order to be compatible with the Arden Syntax, the GLIF Expression Language (GEL) also defines a number of operators that are defined in the Arden Syntax such as '*before*', '*after*' and '*ago*'. Similar constructs are also available in the PRO*forma* expression language. EON expresses criteria using a description that is very similar to that of GLIF, with the main exception that GLIF describes expressions in GEL/GELLO while EON describes expressions by means of the three different criterion languages. The Arden Syntax and GLIF support a limited form of uncertainty in terms of a three-valued logic ('*true*', '*false*' and '*unknown*'). PRO*forma* is the only approach that contains expressive constructs for describing uncertainty aspects of a guideline. As many guidelines (especially treatment guidelines) are rather deterministic by nature, the issue of representing temporal aspects seems to have higher priority that the issue of representation of uncertainty (although this might be less true for diagnostic guidelines).

All approaches have developed execution engines in which the different procedural aspects of the guideline are encoded programmatically (e.g., a number of Java or C procedures that each executes a certain task). The Arden syntax, PRO*forma* and EON have published results on the development and implementation of actual decision support systems in daily care. PRO*forma* is the only approach described here that has developed a commercialised version. PRO*forma*, EON and Asbru execution engines are able to communicate with clinical information systems and users through standard Application Programming Interfaces (APIs) or communication protocols (e.g., web services).

A number of third parties have implemented decision support systems that are able to execute Arden Syntax guidelines for use in their local institutions. However, these are often not reusable in other environments.

6.2. Requirements

The descriptions and comparisons in the previous sections show that each approach has a number of strong points. This section formulates requirements that were distilled from these points that can be used in the process of developing new approaches or improving existing ones.

6.2.1. Model

A guideline TNM must contain a set of generic guideline tasks that is able to represent all facets of simple as well as complex diagnostic and treatment guidelines. This set must be understandable on a functional level by guideline authors and on an executable level by computerised decision support systems.

A guideline TNM must support at least the two necessary basic tasks: actions and decisions. In order to be able to specify guideline-oriented actions (e.g., '*prescribe new medication*' or '*diagnose patient with hypertension*') as well as programming-oriented actions (e.g., '*get all drugs from an EPR*' or '*give message to user*') a guideline TNM must 1) provide a very expressive and rich model that enables the specification of all above-mentioned actions in a limited set of tasks or 2) provide the ability to derive new (sub)tasks from the existing ones that define new functionality.

Other important tasks in a guideline TNM are tasks that influence guideline flow such as entry/exit points (e.g., *Patient state* steps) and repetition/loops (e.g., *Synchronization* steps or the Asbru *Plan type*).

The guideline TNM must be able to represent various kinds of guidelines, that may differ considerably in complexity in a consistent manner such as relatively simple guidelines that model independent modular rules (e.g., MLMs in the Arden Syntax or MLM-macros in GLIF), but also complex guidelines such as clinical trials or treatment plans. In order to represent these various types of guidelines in a consistent manner, the approach must be able to represent guidelines on multiple levels of abstraction such as nesting, task or guideline decomposition (e.g., subguidelines or subplans in GLIF, EON, PRO*forma* and Asbru), and specifying the guideline's intention or goal (e.g., Asbru's intentions).

CIGs that are used for active decision support must be integrated with existing clinical information systems such as EPR systems. Concepts that are used in a guideline such as patient demographics, results of laboratory tests, indications and drugs must be explicitly defined so that they can be mapped to entries in a clinical information system. To facilitate the (re)use of a guideline among different institutions and systems, the reasoning knowledge (e.g., the used methods or tasks) must be separated from domain-specific knowledge (e.g., used drugs or laboratory tests). Also, the representation should support the use of standard data models and medical terminologies such as HL7, UMLS and SNOMED (e.g., the multi-level approaches in GLIF and EON).

Furthermore, in order to further facilitate the sharing of guideline-based decision support systems and to increase the acceptance of (national) guidelines in local institutions, actions that are programming-related must be separated from actions that are not. In this manner, institution-specific actions (e.g., sending an email to a physician vs. showing a message on a screen) are defined separate from the knowledge that describes the guideline itself. For example, guidelines may contain an additional 'layer' that describes such actions, independent of the guideline process. This is supported by GLIF and EON as it is possible to describe multiple kinds of tasks for each action such as decision support-related or programming-related tasks.

A guideline representation must be able to hold didactic- and maintenance-related information such as author names, versions, (literature) references, sources and referees. Especially versioning-related information is very important, as guidelines are usually dynamic (the contents may change rapidly over time) and national guidelines may be adapted to local institutions.

6.2.2. Language

A CIG approach should define formal control-flow languages as well as expression languages that are able to capture all the requirements mentioned above, in an unambiguous way. On the one hand, these languages must be abstract enough so that it is interpretable by guideline acquisition/visualisation tools (e.g., Protégé, Arezzo Composer) and guideline authors who do not have a logical or modelling background are able to define the process (e.g., flow), decision criteria and actions in a guideline (a more detailed description on various guideline acquisition/visualisation tools can be found elsewhere [3] and in Chapter 8: Visualization Methods to Support Guideline-Based Care Management). On the other hand, the languages must be interpretable (and preferable also verifiable) by engines that are able to execute guidelines. Such an execution engine must be able to interface with various clinical information systems in a consistent manner, for example by mapping concepts from the guideline to corresponding items in a clinical information system (e.g., the concept *Drug* in a guideline must be mapped to a drug table of an information system's database). Also, actions that a guideline performs must be configurable as they may differ in various local situations (e.g., send an e-mail in a certain situation in contrast to issuing an on-screen alert in another one). This implies a component-based approach in which each component performs a specific task such as reasoning or interfacing. The encoded format as well as the guideline execution engine must meet execution-time requirements such as compactness and execution speed.

Temporal logic is a very important issue in guideline modelling. Guidelines usually refer to complex temporal constructs to describe for example drug prescription schemes. Therefore, a guideline representation model must contain an expressive means of modelling temporal expressions (e.g., Asbru's temporal logic). The truth-value of a decision can not always be evaluated as '*true*' or '*false*', for example in the case of missing data (e.g., the patient's medical history is not known). Guideline models must be able to handle such situations (e.g., using the relatively simple three-valued logic in GLIF or the more complex R^2L language in PRO*forma*).

7. Research Agenda

In the last decade, most of the attention on computer-based guideline development has been focused on the areas of guideline representation models and underlying languages. From this research, various approaches and models arose, each with its own focus points and related strengths and weaknesses. However, based on the comparison in this chapter as well as looking at other studies [3, 4, 5-8], the conclusion can be drawn that the minimal necessary components for guideline representation have been identified. The next step in this process will be to develop a standard guideline representation model using these components (e.g., under the HL7 auspices [54]), keeping in mind the main conclusion that was drawn by Peleg et al.: '*…that because of the different goals of various research groups, a consensus model will be acceptable to the research groups only if it concurrently allows them to continue their investigations of unique features*' [8].

Also, the real benefit lays in structuring and guiding the whole guideline development process: in order to successively computer-based guideline systems that will be used in daily practice, various aspects such as representation, acquisition, verification and execution must be taken into account (a nice example of such an approach is the DeGel project from the Ben-Gurion university [47]). This is not a trivial task. Comparing the various approaches shows that design specifications made in the area of guideline representation have implications in the area of guideline execution (e.g., the 'fuller' the language, the less executable it will be).

Although significant progress has been made during the last years, especially regarding guideline representation, several issues that relate to guideline implementation and guideline-based decision support still have to be addressed more extensively. Examples of such issues are how to implement national guidelines as well as local adaptations of those guidelines and how to increase the shareability of generic guideline execution engines among different intuitions (e.g., the GESDOR approach). Various solutions may be developed that address these issues such as the development of versioning methods that enable synchronisation between national and local guidelines and the development of standard interfaces to different external information systems. Recently, a number of articles have been published that compare CIG approaches with traditional workflow languages [6]. We expect that these comparisons will contribute significantly to the process of defining requirements and standards for CIG approaches (see also Chapter 3: From guidelines to careflows: modelling and supporting complex clinical processes).

In order to create an approach that is successful, it is important that future research will take into account that an acceptable compromise between all areas must be reached with the above-mentioned aspects as starting points. In this compromise, a balance must be maintained between the aspects of abstractness, expressiveness, formalisation, acquisition and execution.

References

[1] James BC. Making it easy to do it right. N Engl J Med 2001;345(13):991-3.

[2] OpenClinical. Available at: http://www.openclinical.org. Accessed March 11, 2008.

[3] De Clercq PA, Blom JA, Korsten HH, Hasman A. Approaches for creating computer-interpretable guidelines that facilitate decision support. Artif Intell Med. 2004 May;31(1):1-27.

[4] Peleg M. Guideline and Workflow Models. In: Medical Decision-Making: Computational Approaches to Achieving Healthcare Quality and Safety, Robert A. Greenes (ed.), Elsevier/Academic Press, 2006.

[5] Tu S, Musen M. Representation Formalisms and Computational Methods for Modeling Guideline-Based Patient Care. First European Workshop on Computer-based Support for Clinical Guidelines and Protocols 2000;:125-42.

[6] Mulyar N, van der Aalst WM, Peleg M. A pattern-based analysis of clinical computer-interpretable guideline modeling languages. J Am Med Inform Assoc. 2007 Nov-Dec;14(6):781-7.

[7] Wang D, Peleg M, Tu SW, Boxwala AA, Greenes RA, Patel VL, Shortliffe EH. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. Int J Med Inform. 2002 Dec 18;68(1-3):59-70.

[8] Peleg M, Tu S, Bury J, Ciccarese P, Fox J, Greenes RA, Hall R, Johnson PD, Jones N, Kumar A, Miksch S, Quaglini S, Seyfang A, Shortliffe EH, Stefanelli M. Comparing computer-interpretable guideline models: a case-study approach. J Am Med Inform Assoc. 2003 Jan-Feb; 10(1): 52-68.

[9] De Clercq PA, Hasman A. Experiences with the development, implementation and evaluation of automated decision support systems. Medinfo 2004;11(Pt 2):1033-7.

[10] Elkin PL, Peleg M, Lacson R, Bernstam E, Tu S, Boxwala A, Greenes R, Shortliffe EH. Toward Standardization of Electronic Guideline Representation. MD Computing, 2000 17(6):39-44.

[11] Clayton PD, Pryor TA, Wigertz OB, Hripcsak G. Issues and structures for sharing knowledge among decision-making systems: The 1989 Arden Homestead Retreat. In: Kingsland LC (ed). Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care. New York: IEEE Computer Society Press. 1989;:116-21.

[12] Ohno-Machado L, Gennari JH, Murphy SN, Jain NL, Tu SW, Oliver DE. Pattison-Gordon E. Greenes RA. Shortliffe EH. Barnett GO. The guideline interchange format: a model for representing guidelines. JAMIA 1998;5(4):357-72.

[13] Fox J, Johns N, Rahmanzadeh A. Disseminating medical knowledge: the PRO*forma* approach. Artif Intell Med 1998;14:157-81.

[14] Shahar Y, Miksch S, Johnson P. The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. Artif Intell Med 1998;14:29-51.

[15] Musen MA, Tu SW, Das A, Shahar Y. EON: A Component-Based Approach to Automation of Protocol-Directed Therapy. JAMIA 1996;3:367-88.

[16] Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res 1994;27(4):291-324.

[17] Peleg M, Boxwala AA, Bernstam E, Tu S, Greenes RA, Shortliffe EH. Sharable representation of clinical guidelines in GLIF: relationship to the Arden Syntax. J Biomed Inform. 2001 Jun; 34(3):170-81.

[18] Clinical Decision Support & Arden Syntax Technical Committee of HL7, inventor Arden Syntax for Medical Logic Systems, version 2.0. Draft revision. USA. July 7, 1999.

[19] Pryor TA, Hripcsak G. Sharing MLMs: an experiment between Columbia-Presbyterian and LDS Hospital. Proc Annu Symp Comput Appl Med Care 1993;:399-403.

[20] Sherman EH, Hripcsak G, Starren J, Jenders RA, Clayton P. Using intermediate states to improve the ability of the Arden Syntax to implement care plans and reuse knowledge. Proc Annu Symp Comput Appl Med Care 1995;:238-42.

[21] Hripcsak G, Cimino JJ, Johnson SB, Clayton PD. The Columbia-Presbyterian Medical Center decision-support system as a model for implementing the Arden Syntax. Proc Annu Symp Comput Appl Med Care 1991;:248-52.

[22] Kuhn RA, Reider RS. A C++ framework for developing Medical Logic Modules and an Arden Syntax compiler. Comput Biol Med 1994;24(5):365-70.

[23] McCauley B, Young I, Clark I, Peters M. Incorporation of the Arden Syntax within the reimplementation of a closed-loop decision support system. Comput Biomed Res. 1996 Dec;29(6):507-18.

[24] Noy NF, Crubezy M, Fergerson RW, Knublauch H, Tu SW, Vendetti J, Musen MA. Protégé-2000: an open-source ontology-development and knowledge-acquisition environment. AMIA Annu Symp Proc. 2003;:953.

[25] Peleg M, Boxwala AA, Tu S, Zeng Q, Ogunyemi O, Wang D, Patel VL, Greenes RA, Shortliffe EH. The InterMed approach to sharable computer-interpretable guidelines: a review. J Am Med Inform Assoc. 2004 Jan-Feb;11(1):1-10.

[26] Boxwala AA, Peleg M, Tu S, Ogunyemi O, Zeng QT, Wang D, Patel VL, Greenes RA, Shortliffe EH. GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. J Biomed Inform. 2004 Jun;37(3):147-61.

[27] Choi J, Currie LM, Wang D, Bakken S. Encoding a clinical practice guideline using guideline interchange format: A case study of a depression screening and management guideline. Int J Med Inform. 2007 Oct;76 Suppl 2:S302-7.

[28] Wang D. Translating Arden MlMs into GLIF guidelines--a case study of hyperkalemia patient screening. Stud Health Technol Inform. 2004;101:177-81.

[29] Schadow G, Russler DC, Mead CN, McDonald CJ. Integrating Medical Information and Knowledge in the HL7 RIM. Proc AMIA Symp 2000;:764-8.

[30] Lindberg C. The Unified Medical Language System (UMLS) of the National Library of Medicine. J Am Med Rec Assoc 1990;61(5):40-2.

[31] Peleg M, Keren S, Denekamp Y. Mapping computerized clinical guidelines to electronic medical records: Knowledge-data ontological mapper (KDOM). J Biomed Inform. 2008 Feb;41(1):180-201.

[32] Peleg M, Ogunyemi O, Tu SW, Boxwala AA. Zeng Q. Greenes RA. Shortliffe EH. Using Features of Arden Syntax with Object-Oriented Medical Data Models for Guideline Modeling. Proc AMIA Symp 2001;:523-7.

[33] Sordo M, Ogunyemi O, Boxwala AA, Greenes RA. GELLO: An Object-oriented Query and Expression Language for Clinical Decision Support. Proc AMIA Symp. 2003; : 1012.

[34] Wang D, Shortliffe EH. Design and implementation of the GLIF3 guideline execution engine. J Biomed Inform. 2004 Oct;37(5):305-18.

[35] Wang D, Peleg M, Bu D, Cantor M, Landesberg G, Lunenfeld E, Tu SW, Kaiser GE, Hripcsak G, Patel VL, Shortliffe EH. GESDOR – a generic execution model for sharing of computer-interpretable clinical practice guidelines. Proc AMIA Annual Symp 2003;:694-698.

[36] InferMed Ltd. Available at http://www.infermed.com. Accessed March 11, 2008.

[37] The Tallis Toolset. Available at http://www.acl.icnet.uk/lab/tallis/index.html. Accessed March 11, 2008.

[38] Fox J, Patkar V, Thomson R. Decision support for health care: the PROforma evidence base. Inform Prim Care. 2006;14(1):49-54.

[39] Fox J, Das S. Arguments about beliefs and Actions: Decision making in the Real World. In: Fox J, Das S. Safe and Sound: Artificial Intelligence in Hazardous Applications 2000: 55-76.

[40] Fox J, Das S. The RED Knowledge Representation Language. In: Fox J, Das S. Safe and Sound: Artificial Intelligence in Hazardous Applications 2000: 191-206.

[41] Asbru syntax. Available at http://www.asgaard.tuwien.ac.at/plan_representation/asbrusyntax.html. Accessed March 11, 2008.

[42] Seyfang A, Miksch S, Marcos M, Wittenberg J, Polo-Conde C, Rosenbrand K. Bridging the Gap between Informal and Formal Guideline Representations. Proc. 17th European Conference on Artificial Intelligence, 2006;:447-51.

[43] Votruba P, Seyfang A, Paesold M, Miksch A. Improving the Execution of Clinical Guidelines and Temporal Data Abstraction in High-Frequency Domains, AI Techniques in Healthcare: Evidence-based Guidelines and Protocols, European Coordinating Committee for Artificial Intelligence 2006;:112-6.

[44] Protocure II. Available at http://www.protocure.org. Accessed March 11, 2008.

[45] Miksch S, Fuchsberger C. Asbru's Execution Engine: Utilizing Guidelines for Artificial Ventilation of Newborn Infants. Proc. IDAMAP 2003; S:119-25.

[46] Young O, Shahar Y, Liel Y, Lunenfeld E, Bar G, Shalom E, Martins SB, Vaszar LT, Marom T, Goldstein MK. Runtime application of Hybrid-Asbru clinical guidelines. J Biomed Inform. 2007 Oct;40(5):507-26.

[47] Shahar Y, Young O, Shalom E, Galperin M, Mayaffit A, Moskovitch R, Hessing A. A framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools. J Biomed Inform. 2004 Oct;37(5):325-44.

[48] Tu SW, Musen MA. A flexible approach to guideline modeling. Proc AMIA Symp 1999;:420-4.

[49] Tu SW, Campbell JR, Glasgow J, Nyman MA, McClure R, McClay J, Parker C, Hrabak KM, Berg D, Weida T, Mansfield JG, Musen MA, Abarbanel RM. The SAGE Guideline Model: achievements and overview. J Am Med Informtgt Assoc. 2007 Sep-Oct;14(5):589-98.

[50] Goldstein MK, Coleman RW, Tu SW, Shankar RD, O'Connor MJ, Musen MA, Martins SB, Lavori PW, Shlipak MG, Oddone E, Advani AA, Gholami P, Hoffman BB. Translating research into practice: organizational issues in implementing automated decision support for hypertension in three medical centers. J Am Med Inform Assoc. 2004 Sep-Oct;11(5):368-76.

[51] Johnson PD, Tu S, Booth N, Sugden B, Purves IN. Using scenarios in chronic disease management guidelines for primary care. Proc AMIA Symp. 2000;:389-93.

[52] Tu SW, Musen MA. Modeling Data and Knowledge in the EON Guideline Architecture. Medinfo 2001;10:280-4.

[53] Bury JP, Saha V, Fox J. Supporting ‘scenarios’ in the PROforma guideline modeling format. Proc AMIA Symp 2001;:870-4.

[54] Jenders RA, Sailors RM. Convergence on a standard for representing clinical guidelines: work in health level seven. Medinfo 2004;11(Pt 1):130-4.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008
© 2008 The authors and IOS Press. All rights reserved.
doi:10.3233/978-1-58603-873-1-44

Chapter 3
From Guidelines to Careflows: Modelling and Supporting Complex Clinical Processes

John FOX[1,a,b], Elizabeth BLACK[a], Ioannis CHRONAKIS [a,b], Robert DUNLOP [c], Vivek PATKAR [b], Matthew SOUTH[a], Richard THOMSON[d]
[a] Department of Engineering Science, University of Oxford, Oxford, UK
[b] UCL Medical School and NHS Royal Free Hospital, London UK
[c] InferMed Ltd, London, UK; [d] OpenClinical

Abstract. Research on computer interpretable clinical guidelines has largely focused on individual points of care rather than processes of care. Whether we consider simple aids like clinical alerts and reminders or more sophisticated data interpretation and decision-making, guideline developers tend to focus on specific tasks rather than processes like care plans and pathways which are extended in time. In contrast, research on business process modelling has demonstrated notations and tools which deal directly with process modelling, but has not been concerned with problems like data interpretation and decision making. In this chapter we describe these two traditions, and compare some of their strengths and weaknesses. We also briefly discuss the distinct theoretical frameworks which have grown up around them, notably Petri nets for workflow modelling and mathematical logics for guidelines. We conclude that these offer complementary views of clinical processes and that a key research challenge is find a way of unifying them. [2]

Keywords: clinical decision-making, process modelling, workflow languages, task network languages.

Introduction

A standard definition of clinical practice guidelines (CPGs) is that of Field and Lohr [6]: "systematically developed statements to assist practitioners and patient decisions about appropriate health care for specific circumstances". A primary purpose of CPGs is to support clinical decision-making in a way that is consistent with published and peer-reviewed evidence, in order to:

- provide a more rational basis for decision-making;

[1] Corresponding Author: John Fox, Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK; E-mail: johnfox@robots.ox.ac.uk

[2] We wish to thank Natalie Mulyar, Marco Montali, Annette ten Teije and Mor Peleg for their helpful comments on earlier drafts of this discussion.

- provide a focus for continuing education;
- reduce inappropriate variation in practice;
- facilitate clinical audit;
- promote efficient use of resources.

CPGs usually take the form of text documents, sometimes augmented with more structured information such as "clinical algorithms" (flow diagrams). They are usually designed to inform clinicians about best practice and encourage changes in their decision-making where necessary.

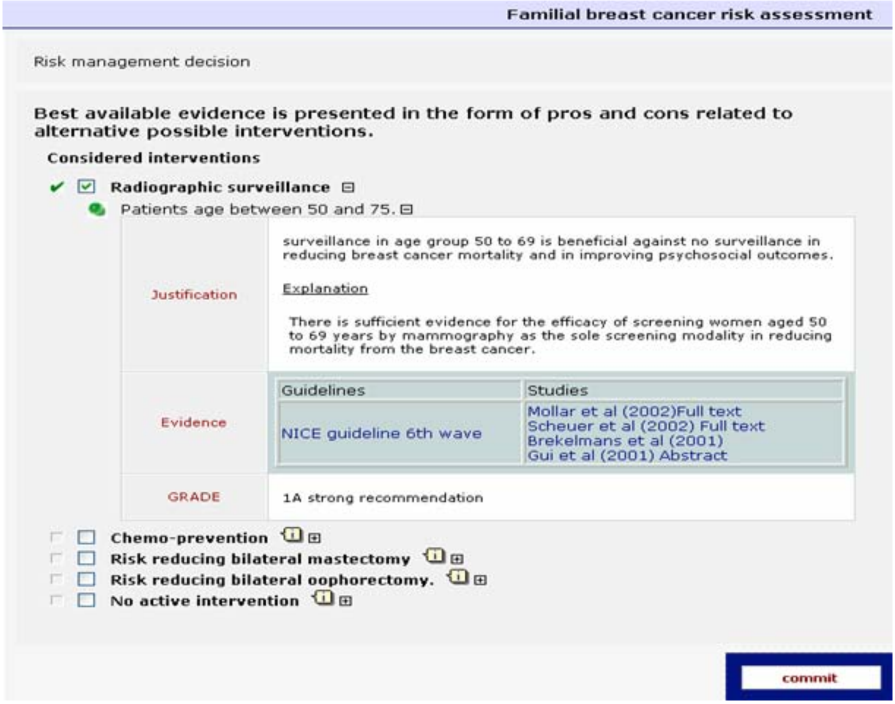
There are, however, significant issues about the effectiveness of CPGs, including doubts about their effectiveness in changing the behaviour of health care professionals [12]. One of the most consistent findings in health services research is the gap between research evidence and routine patient care [15] and the effort that goes into creating the guidelines may not be matched by the level of adherence to them in practice [3].

The medical informatics community has therefore sought new ways of bringing up-to-date scientific and clinical knowledge to the point of care in a more flexible and usable form than human-readable documents. "Computer-Interpretable Guidelines" (CIGs) are formal representations of CPGs that can be used to provide active support for improved effectiveness and safety of clinical practice. CIGs can provide reminders and alerts, assess individual risks, recommend possible treatments and give other patient-specific advice, often providing direct links to the supporting research and evidence as part of the advice. Figure 1 shows an example of a CIG for assessing the genetic risk of breast cancer and suggesting appropriate pre-emptive interventions (e.g. surveillance, chemo-prevention or surgery). It is now technically practical to deliver such services in the doctor's office, on a bedside computer or by PDA.

The creation and use of CIGs have many potential and demonstrated benefits, including:

- offering better description and recording of patient states;
- providing selective access to background knowledge which is relevant to the specific circumstances;
- automatically proposing timely reminders and making patient-specific recommendations for clinical decisions;
- providing the rationale for recommendations, e.g. decision criteria and justifying evidence, and
- facilitating use of formal verification and other quality management techniques (e.g. http://www.protocure.org/)

Recent systematic reviews by Garg et al [13], Kawamoto [16] and others clearly show that CIGs have clinical value. Although these reviews do not cover all published work they provide an interesting insight into the current state of CIG research. Table 1 is a summary and reclassification of the systems considered in the Garg et al review carried out to bring out the distinct kinds of decision support which all come under the general heading of CIGs.



Familial breast cancer risk assessment

Risk management decision

Best available evidence is presented in the form of pros and cons related to alternative possible interventions.

Considered interventions

✔ ☑ **Radiographic surveillance** ⊟

● Patients age between 50 and 75. ⊟

Justification	surveillance in age group 50 to 69 is beneficial against no surveillance in reducing breast cancer mortality and in improving psychosocial outcomes. Explanation There is sufficient evidence for the efficacy of screening women aged 50 to 69 years by mammography as the sole screening modality in reducing mortality from the breast cancer.
Evidence	Guidelines: NICE guideline 6th wave Studies: Mollar et al (2002)Full text Scheuer et al (2002) Full text Brekelmans et al (2001) Gui et al (2001) Abstract
GRADE	1A strong recommendation

□ **Chemo-prevention** ⓘ ⊞

□ **Risk reducing bilateral mastectomy** ⓘ ⊞

□ **Risk reducing bilateral oophorectomy.** ⓘ ⊞

□ **No active intervention** ⓘ ⊞

Figure 1: Example of clinical advice in managing women at low genetic risk of breast cancer. The system considers all the possible interventions. In this simple example the advice is for routine X-ray surveillance, together with the justification, evidence and strength of the recommendation (central panel). Most decision support systems are similarly focused on individual decisions rather than complex processes of care which may be carried out over days or even months.

Table 1: a selective classification of different types of clinical decision support systems reviewed by Garg et al, 2005. Column 3 shows the number of each type and column 4 the number of these which demonstrated significant clinical benefits.

Service type	Example techniques	Instances	
Monitoring, alerts and reminders	Algorithmic and rule based methods	41	30
Modelling and prediction	Calculators, Statistical modelling	35	24
Focusing and information retrieval	Search engines, navigation, InfoButtons	11	6
Framing and making decisions	Decision analysis, logical decision models, argumentation	7	2
Support for complex and multidisciplinary care	Workflow	0	0

A striking feature of table 1 is that the vast majority of evaluations to date deal with clinical systems which are limited to a single point in time where data need to be recorded, alerts flagged or decisions made as in figure 1. There have been few studies of how to integrate CIGs into care planning or clinical workflow, and none in the Garg sample. However with the apparent success of simple CIG technologies the research community is turning its attention to more ambitious goals. Examples include work on decision support within a typical workflow for assessment of women with suspected breast cancer (Patkar et al, this volume) and work on risk assessment and care planning by Glasspool et al [14]. The *1st International Workshop on Process-oriented Information Systems in Healthcare* took place in Brisbane, Australia in 2007, where central themes of the meeting were the use of guidelines in the context of business process and workflow models.

The next section reviews a number of approaches to the challenge of integrating decision support into clinical process models. The review is not comprehensive, providing only a short summary of early approaches to process modeling, but it provides a historical context for discussing more recent technologies. The primary focus of the discussion is how CIG technologies might benefit from work on business processes, and in particular whether a class of CIG technology called *Task Network Languages* can provide effective decision support within the practical constraints of clinical process management.

1. Modelling and Supporting Business Process

Despite the recent growth of interest in process modelling the history of the subject goes back almost half a century[3].

1.1 Critical Path Analysis

The critical path method was developed for scheduling a set of activities in any project with interdependent activities. The essential technique is to develop a model of all the activities required to complete the project, the time that each activity will take to completion, and the dependencies between the activities. CPM was developed in the 1950s, and has been used in construction, software development, research project management, product development, engineering, and plant maintenance, among other applications.

1.2 Gantt Charts

A Gantt chart illustrates a project schedule, showing the start/finish dates of the component tasks of a project aligned on a timeline and showing the status of planned and active tasks (figure 2 (a)). Some Gantt charts also show dependencies between tasks (e.g. preconditions on initiating a task which depend on the completion of another activity or delivery of an output).

[3] In preparing this section we have drawn significantly on material on Wikipedia (www.wikipedia.org). The original articles are far more extensive and are recommended for readers in finding out more detail about the topics.

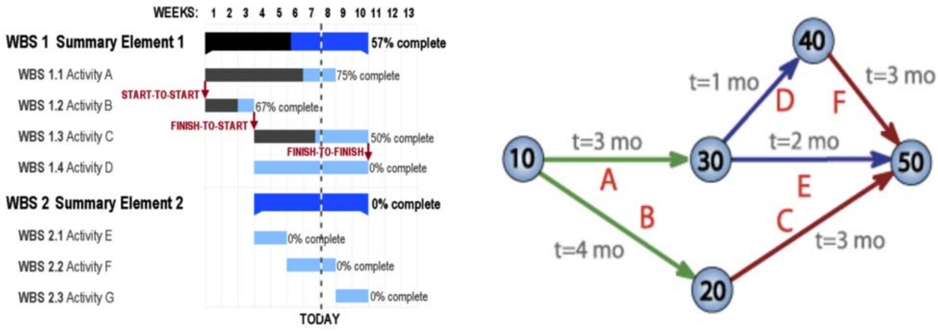


Figure 2: (a) Gantt chard for a simple project of 2 main tasks plus component activities. (b) PERT chart for a seven-month project with five milestones, 10 through 50m and six activities, A through F [source *Wikipedia*].

1.3 PERT Charts

PERT was invented as a technique for managing large military projects to simplify planning and scheduling and is commonly used in R&D-type projects where time, rather than cost, is the major factor. The distinctive feature of PERT is the chart of interconnecting timelines, such as the one shown in figure 2(b). Unlike Gantt charts the PERT method accommodates some uncertainty by making it possible to schedule a project while not knowing precisely the details and durations of all the activities.

These early methods for modelling business processes have considerable generality and practical value. However, they also have limitations from the point of view of clinical process management. They were not developed for actively executing or supporting the management of the process in real time, which is a key objective of CIGs. Neither are they oriented towards personalisation of care plans nor dynamic modifications of treatment plans in the light of changing situations or unexpected events. They were designed primarily for managing one-off projects, not for supporting routine processes like clinical diagnosis and treatment, and are typically used for analysing dependencies in a process and detect overruns or other problems.

1.4 Petri Nets

Petri nets can be used for describing “systems that may be concurrent, asynchronous, distributed, parallel, nondeterministic and/or stochastic” and are an advance on the earlier static methods in a number of ways. Petri nets provide a formal representation of a process as a directed graph with annotations, which can be interpreted dynamically to simulate or “enact” that process. There are two kinds of nodes in a Petri net: *places* and *transitions*. Places represent conditions and transitions represent events. A transition node has a certain number of input and output places representing the preconditions and post-conditions of the event. Petri nets are a very general model which can capture Gantt and PERT charts as well as finite state machines, dataflow networks and in theory any process that can be represented as a graph. The basic abstraction (*input places - transitions - output places*) subsumes many different concrete forms, including (*input data – computation – output data*); (*resources needed – task or job – resources released*); (*preconditions – event – post-conditions*) and other instances of the abstraction.

Petri Nets are often said to lack compositionality and scalability. These problems have been addressed by “higher-level Petri nets”, such as *Object-Oriented Petri Nets*. In a higher-level net, a Petri Net (PN) can appear wherever a data object can appear: as a token, as a parameter, or as the value of a computation. Tokens may carry arbitrary data objects and functions. Object-Oriented concepts have been used to build “layered" Petri Nets and many different languages have been created to integrate OO concepts into PN modelling. PNs have also been used to provide a foundation for the most recent tradition for modelling processes, workflow modelling, in which concepts like processes, activities and messages may be modelled as a layer of Petri net objects on top of the basic PN primitives.

1.5 Rule-based Systems

A prominent attempt to develop a standard for modelling clinical guidelines and decision support services was the Arden Syntax for Medical Logic Systems, developed in the early nineties. Arden Syntax encodes medical knowledge as Medical Logic Modules (MLMs), a hybrid between a production rule (an "if-then" rule) and a procedural formalism. Each MLM is invoked as if it were a single-step "if-then" rule, but then it executes serially as a sequence of instructions, including queries, calculations, logic statements and write statements. MLMs have been used to generate clinical alerts and reminders, interpretations, diagnoses, screening for clinical research studies, quality assurance functions, and administrative support. Arden was initially designed to support clinical decision making (an individual MLM should contain sufficient logic to make a single medical decision) but it can capture simple task sequences by chaining MLMs.[4]

Although Arden was adopted as an ANSI standard for computerised clinical guidelines it lacks expressivity for modelling clinical guidelines (e.g. for logical reasoning, representing time and uncertainty) and limited ability to represent complex processes (e.g. care pathways and protocols). Other rule-based technologies have considerably more power in these respects. For example rule-based “expert” systems were extensively developed for medical applications in the 70s and 80s (e.g. MYCIN: [22]). These led to two distinct classes of rule-based expert system “shells”, goal-directed and pattern-directed application development tools (e.g. eMYCIN; [24] and PSYCO [7]).

Perhaps the most general and powerful form of rule-based technology is associated with logic programming. Originally developed for natural language processing, logic programming separates the declarative aspects of computation (e.g. what medical knowledge do we have?) from the procedural aspects (how is it to be applied during the clinical process?). Logic programming languages have great expressive power and the clear semantics of first-order logic. Defining process definition languages and implementing specialised process enactment engines for those languages is well within the capabilities of a logic programming language, such as Prolog, and also supports the technique of “meta-logical programming” which facilitates reuse of declarative knowledge and logic programs in different ways and different settings.

[4] Source www.openclinical.org

1.6 Workflow Languages and Technologies

Business process modelling, and associated IT systems for enacting and supporting business processes have attracted great interest in academia and practical commerce in recent years. Figure 3 illustrates the relationship between a business process and two key concepts: the idea of a *process definition language* that formally describes the process model, and the *workflow management system*, the technology platform which supports the management of that process by scheduling activities, managing data, and ensuring effective communication of and synchronising between interrelated or dependent activities. Execution of a particular instance of a business process, such as the management of an order for a product or service, is initiated and controlled by the workflow management system under the control of the process definition for that kind of order.

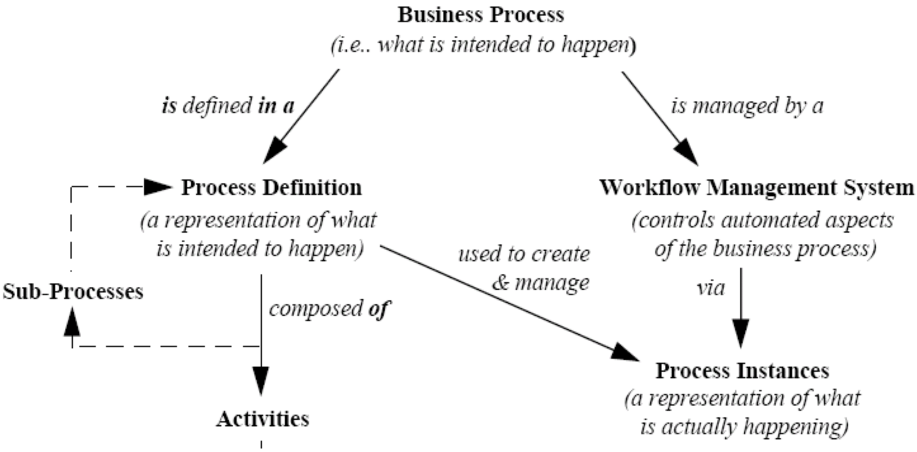

Business Process
(i.e.. what is intended to happen)
is defined in a
is managed by a
Process Definition
(a representation of what is intended to happen)
Workflow Management System
(controls automated aspects of the business process)
used to create & manage
via
Sub-Processes
composed of
Process Instances
(a representation of what is actually happening)
Activities

Figure 3: Business process and workflow modelling: From Workflow Management Coalition glossary. Source http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf

A number of workflow languages have been developed and proposals for standards are now emerging. They have been developed primarily with the business community in mind for modelling and supporting business organisations and processes rather than medical organisations and processes, though the possibility of using them in clinical settings is being actively studied (e.g. [17]).

A variant of workflow called ‘careflow’ has been proposed from within the guidelines community for supporting clinical practice by explicitly modelling guidelines in the context of workflow, communication and resource management services. Originally proposed by Mario Stefanelli and his colleagues at the University of Pavia, careflow is defined as “a [clinical] procedure through which administrative and supervisory tasks … are passed between participants according to a *process definition.* A careflow process definition specifies which *tasks* need to be executed and in which order.” [18]. In his earlier paper Stefanelli suggests that Petri nets are the preferred formalism for modelling careflows but by 2004 the theoretical foundation seems to be in terms of distributed multi-agent systems: a careflow is defined as “the execution of a process definition for a particular patient [by] *organizational agents* [which] … play one or more *roles* … execute work items, use resources (material, instruments, services, etc.) and have a set of rules defining how to *communicate* with other agents in the organization."

1.7 Task Network Languages

During the last decade in which workflow languages have been achieving prominence the medical informatics community has also been developing ways of modelling clinical processes. Although they share concerns with the workflow community the "guidelines community" has taken a distinctive approach. This has been partly because of the original focus on clinical decision-making about how to interpret situations and events (e.g. is the patient at risk? what is the cause of a clinical problem?) or the best choice of clinical action (do we need to carry out further investigations? what is the most effective and safest thing to do?). Furthermore the guidelines community has been concerned to ensure that the process definition formalises processes in a way that reflects clinical tasks and constraints as clinicians perceive these. "Task Network Languages" have emerged as a way of achieving these objectives.

The motivation for developing task network languages for CIGs is summarised by Peleg et al [20]: "A key component of guideline-based decision-support tools is a machine interpretable guideline model … that unfolds over time. To model such guidelines, many groups have adopted an approach involving hierarchical decomposition of a guideline plan into a network of component tasks. … The plan components provide design primitives that are custom-tailored to clinical guidelines. Relationships among plan components, such as ordering constraints, can be described, and tools for the visual representation of plans and the organization of tasks within them are typically provided. This organization of a guideline model is very different from that of rule-based systems, where the flow of control is not explicitly modelled. TNLs differ in their emphases and in their approaches to addressing particular modeling challenges. A subset of them (ASBRU, GLIF3 and PRO*forma*) have put a major emphasis on defining process definition *languages* … which makes them more directly comparable with business process modelling and workflow languages"

Peleg et al reviewed six task network models which had been developed to capture clinical guidelines and evidence-based practice in a computer-executable and interactive form (Asbru, EON, GLIF, GUIDE, PRODIGY, and PRO*forma)* in order to compare similarities and differences and establish a consensus on a set of common components. A unifying feature of the task network schemes is that they represent clinical guidelines in the form of a *guideline plan.* The plan's components represent decisions, actions, or hierarchically decomposed sub-plans of the guideline and their relationships.

The Peleg et al study examined "several aspects of plan organization: (a) supporting computational models, (b) mechanisms for nesting a high-level plan into a network of lower-level plans, (c) sequential execution of plan components, (d) parallel execution of plan components, (e) cyclical and iterative plans, and (f) entry points into guideline plans. Note that sequential-, parallel-, and cyclical-execution define part of the control flow of guideline plans. The other part is modelled by decision models … that conditionally direct control flow into selected branches of the guideline model." Some of these models have led to the development of formal process definition languages that can support data interpretation, alerts, decision-making and the other services required for modelling clinical guidelines (Asbru, GLIF, and PRO*forma*).

Although developed largely independently of commercial business process engineering TNLs share some features with business process modelling and workflow languages. In the next section we will compare and contrast TNLs and workflow languages as candidates for developing a generalised approach.

2. Comparison of Workflow and Task Network Languages

The purpose of this section is to explore whether guideline technologies in the form of TNLs offer an adequate foundation for a general representation language and implementation platform for clinical process management systems. We shall suggest that as well as supporting decision-making and a "natural" conceptualisation of clinical processes TNLs can capture many workflow patterns required in typical care plans. To illustrate this we will compare one particular workflow language, Business Process Modelling Notation (BPMN), with one task network language, PRO*forma*, developed in our group. This has been used in many clinical applications (e.g. [10]). PRO*forma* has also been the subject of independent comparisons with other TNLs ([5, 20]) and workflow languages ([2[5];20].. However, we only wish to use it here as an exemplar of the general approach. A similar analysis could be carried out for any of the languages within the general family.

2.1 Workflow Modeling Languages

Business process modelling languages are intended to allow designers to formalise a process definition such as a workflow, in which the notation shows visually who does what, where and in what order activities are carried out. Figure 4 shows an example of a model of a clinical process in one business process notation, BPMN, in which a woman with suspected breast cancer is referred to a specialist breast clinic by a general practitioner (GP). The GP and the clinic are "collaborating businesses" in BPMN terminology. The clinic also collaborates with "laboratories" for specialist services that provide specialist investigations such as imaging (mammography) and fine needle aspiration biopsy (FNA).

The BPMN notation can be translated into a standard XML-based interchange format (XPDL) and an executable process definition such as the Business Process Execution Language (BPEL) developed by the Workflow Management Coalition. BPMN draws strongly on conventions such as those used in flow diagrams for showing the flow of control (though the term is avoided in BPM circles) but it also visualises non-control-flow constructs such as trigger events, delays and messages as well as the structure of a business process at internal and collaborative levels.

BPMN supports four main kinds of construct: flow objects; connecting objects; swim-lanes and artifacts.

Flow objects

An **activity** is any kind of work that the business performs. An activity can be atomic (a task) or non-atomic (a sub-process).

[5] http://www.openclinical.org/docs/ext/briefingpapers/bensonPathways.pdf

Gateways are used to control branching and merging of flows and can be the points at which decisions are taken.
An **event is** anything that can "happen" during the course of a business process which is not under the control of that process. Events usually have a cause (trigger) and/or an effect (result) and can affect the flow of the process.

In figure 4, for example, the start event for the clinical process is that a patient attends a GP clinic, shown as a circle on the boundary of the "GP Community" box; rounded rectangles are internal activities of collaborating organisations, and the diamond is a gateway following completion of diagnosis, when the patient is given a positive diagnosis or the "all clear".

Connecting objects

Flow objects are connected together to model the activity sequences in a business process. Three *Connecting Objects* are supported in BPMN:

Sequence Flow defines the order in which activities are to be performed.
Message Flow is concerned with the flow of information between collaborating businesses or individuals.
Associations are typically used to represent the inputs and outputs of activities in a process.

The black-headed arrows in figure 4 represent sequence flow (the activity at the tail of the arrow must be completed before the activity at the head of the arrow can be started). The dotted arrows with white heads are messages and flows of information to and from the specialist laboratories and between the GP and the clinic.

Swim-lanes

Many process modelling tools use "swim-lanes" to show independent "actors" e.g. businesses or services. BPMN supports two main constructs: pools and lanes.

Pools are used to represent separate business entities and are physically separated in a business process diagram. In Figure 4 there are three pools, one representing the GP, one the breast clinic and the third the specialist labs. The activities within pools are considered to be self-contained processes so that sequence flow may not cross the boundary of a pool. Message flow is used to model the communication between participants so it must connect between two pools (or the objects within the pools).

Lanes are sub-partitions within a pool and are used to organize and categorize activities. In figure 4 there are two lanes within the laboratories pool, one representing a mammography process and the other a cytology service. Lanes are often used to separate the activities associated with a specific company function or role.

Artefacts

Various application-specific "artefacts" can be added to a model including data objects, documentation and annotations but these are ignored here as they are not part of the formal process model itself.

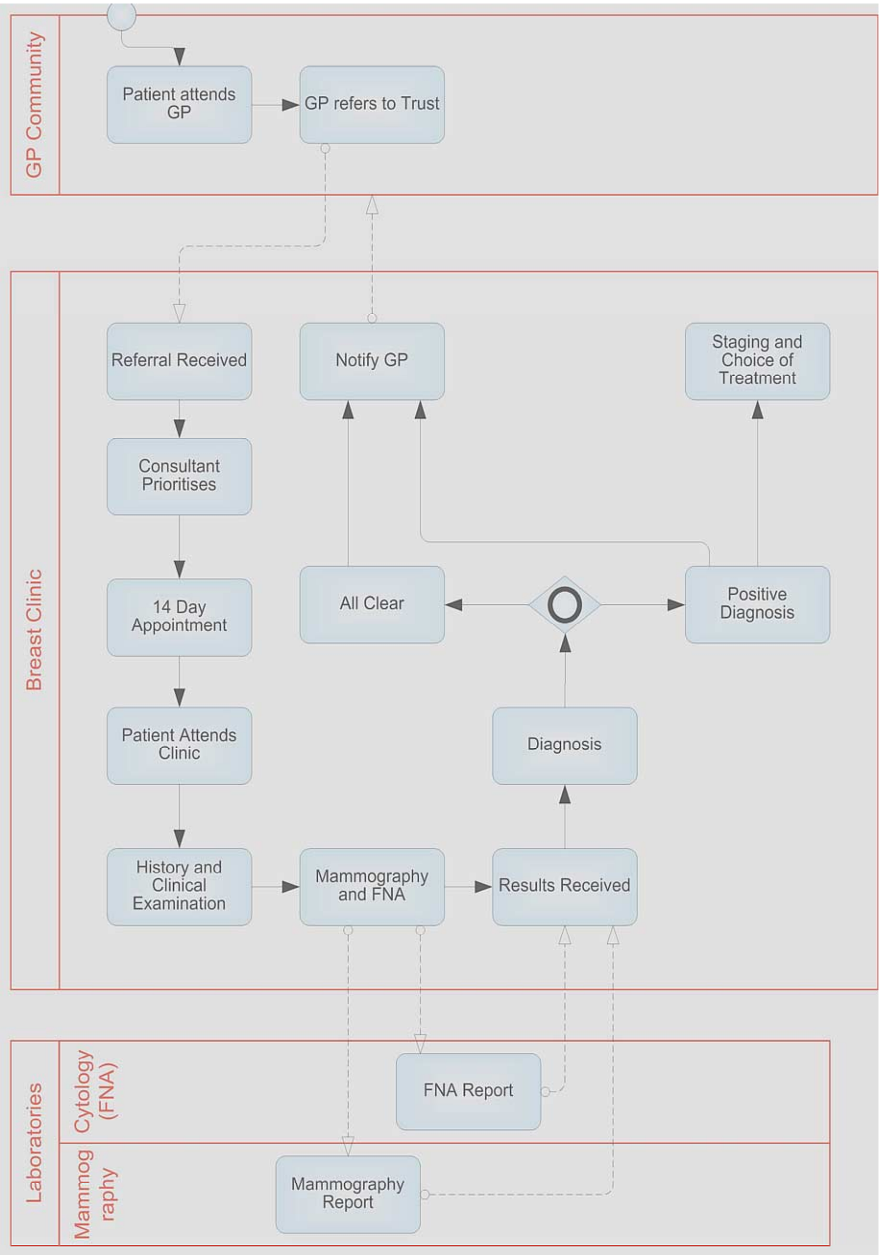


Figure 4: a BPMN model of the management of a patient presenting to her GP with symptoms of possible breast cancer (author Tim Benson, Abies Ltd.).

2.2 Task Network Languages

In common with other TNLs reviewed by Peleg et al PRO*forma* was originally developed as a language to support clinical data interpretation, decision-making and disease management rather than workflow. PRO*forma* is a logic programming language which is extended to support constructs like plans, decisions, data definitions and a number of control mechanisms [4]. It has a declarative syntax and a well-defined operational semantics [23]. PRO*forma* supports the definition of clinical guidelines and protocols in terms of:

- A small set of *task classes* that can be composed into networks representing arbitrarily complex plans or procedures, and a similarly small set of attributes which control task enactment.
- Logical conditions which describe situations, events, constraints, task pre- and post-conditions, and data interpretation rules that influence execution of each task and transfer of control between tasks.

PRO*forma* bases its process model on an ontology of tasks (plans, decisions, actions and enquiries) which is intended to be minimal to keep the language as simple and uniform as possible. All four task types share common attributes which are inherited from the root class (called a "keystone"). All tasks may, for example, have attributes representing the *goal* of the task, *triggering* events, pre-conditions which must be satisfied if the task is to be enacted, and post-conditions which will be true if the task is completed successfully. All tasks have distinguishing attributes as well. *Plans*, for example, have a characteristic structure in terms of nested components (tasks and sub-plans) and plans have special trigger conditions which define when it should be terminated or aborted. *Decisions* are modelled in terms of collections of decision options, logical rules and functions for constructing arguments for and against options, and "commitment" rules. *Actions* and *enquiries* have attributes that support their role in interacting with the external environment.

Figure 5 shows a PRO*forma* process model for the breast cancer referral service modelled using BPMN in figure 4. In this view the GP, clinic and laboratories ("collaborating businesses") are modelled in separate plans in the top left pane of the figure. The internal activities of these businesses (equivalent to "lanes" in BPMN) are also represented as plans, whose internal structure is shown in the 3 panes on the right. Arrows specify "scheduling constraints" which determine that any upstream tasks must be completed before downstream tasks can be initiated. A PRO*forma* engine that enacts these workflows must honour these constraints.

If there are no constraints between tasks then they can be executed concurrently. There are three levels of concurrency in figure 5: the level of the collaborating businesses, the breast clinic (three concurrent workflows) and the laboratories (two concurrent flows). As in the BPMN model concurrent workflows primarily interact by sending messages between activities.

Figure 6 shows a complementary view of the process model which shows the hierarchy of individual tasks that make up the model. Note that although key clinical tasks, like plans, decisions and data capture tasks, are modelled in detail in PRO*forma* there is little support for modelling clinical organisations: there is no specific entity to represent a separate service, for example, nor for modelling communication. In this example plans are used to distinguish services and messages are represented by keystones.

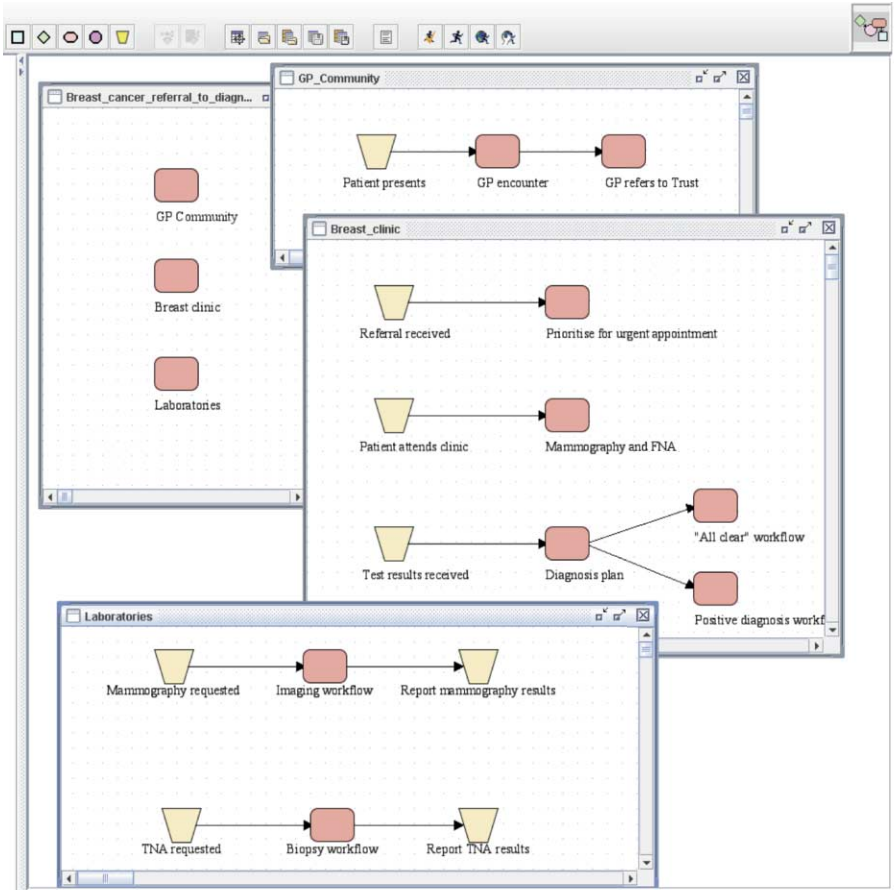
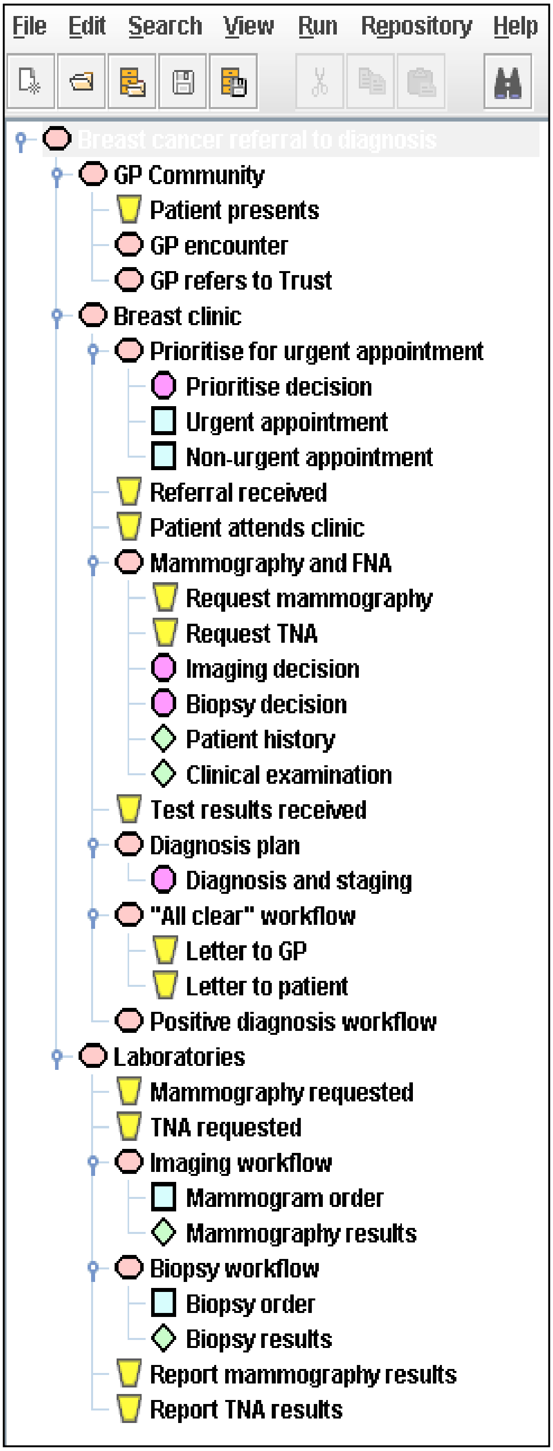


Figure 5: a PRO*forma* version of the BPMN workflow model shown in figure 4.

The absence of formal distinctions between types of swim-lane and lack of communication tasks provided by BPMN suggests that the small PRO*forma* task ontology is being semantically overloaded. This is less serious than it appears. For example, the “businesses”, “pools”, “lanes” iconography could be supported by TNL design tools just as they are by business process modelling tools. A more serious requirement is to extend the task semantics appropriately, which can be done by extending the task ontology.



- Breast cancer referral to diagnosis
  - GP Community
    - Patient presents
    - GP encounter
    - GP refers to Trust
  - Breast clinic
    - Prioritise for urgent appointment
      - Prioritise decision
      - Urgent appointment
      - Non-urgent appointment
    - Referral received
    - Patient attends clinic
    - Mammography and FNA
      - Request mammography
      - Request TNA
      - Imaging decision
      - Biopsy decision
      - Patient history
      - Clinical examination
    - Test results received
    - Diagnosis plan
      - Diagnosis and staging
    - "All clear" workflow
      - Letter to GP
      - Letter to patient
    - Positive diagnosis workflow
  - Laboratories
    - Mammography requested
    - TNA requested
    - Imaging workflow
      - Mammogram order
      - Mammography results
    - Biopsy workflow
      - Biopsy order
      - Biopsy results
    - Report mammography results
    - Report TNA results

Figure 6 PRO*forma* task hierarchy based on the BPMN model in figure 4.

Plans (rounded rectangles) are sets of tasks to be carried out for some (e.g. therapeutic) reason. Like ASBRU and GLIF plans are viewed as the building blocks of a guideline, and may contain any number of tasks of any type, including other plans. Plans may impose an ordering on their components or may permit them to be enacted concurrently..

Decisions (circles) are tasks which involve *choices* of some kind, such as a choice between alternative investigations, diagnoses or treatments. The PRO*forma* specification of a decision task defines the decision options, relevant information, a set of argument rules which evaluate the "pros" and "cons" of each option in a particular setting, an aggregation function which defines a preference ordering on the options, and a commitment rules which determines which of the options should be selected and when.

Actions (squares) are procedures (such as the administration of an injection) which need to be carried out externally to the PRO*forma* process, typically by a clinician.

Enquiries (diamonds) are actions returning required information from the user, EPR or elsewhere.

Keystones are generic tasks, which act as placeholders for specific tasks whose methods may be determined later in the design process, or dynamically at run-time.

A more uncertain issue for TNLs concerns “workflow patterns” which have been developed as standard benchmarks for workflow management systems (see the excellent website created by van der Aalst and ter Hofstede and their colleagues; http://www.workflowpatterns.com/). Mulyar et al [17] have examined four prominent task network formalisms (ASBRU, EON, GLIF and PRO*forma*) from the point of view of whether they can capture 40 benchmark control flow patterns. They found that all TNLs fall significantly short of coping with the set of benchmarks – though it appears that many workflow formalisms are similarly limited![6] However these shortcomings are being addressed by the workflow community. For example YAWL is a new workflow language which has been explicitly designed to cope with a larger set of benchmark patterns, and has a well-defined syntax and semantics based on Petri net formalisms[7].

A more uncertain issue for TNLs concerns “workflow patterns” which have been developed as standard benchmarks for workflow management systems (see the wonderful website created by van der Aalst and ter Hofstede and their colleagues; http://www.workflowpatterns.com/). Mulyar *op cit* have examined four prominent task network formalisms (ASBRU, EON, GLIF and PRO*forma*) from the point of view of whether they can capture 40 benchmark control flow patterns. They found that all TNLs fall significantly short of coping with the set of benchmarks – though it appears that many workflow formalisms are similarly limited![8] However these shortcomings are being addressed by the workflow community. For example YAWL is a new workflow language which has been explicitly designed to cope with a larger set of benchmark patterns, and has a well-defined syntax and semantics based on Petri net formalisms[9].

Comparisons with workflow and other technologies are important in identifying potentially valuable lessons for the guidelines community. However, TNLs may offer techniques that could be useful for enhancing workflow technology. For example PRO*forma* was designed around an explicit model of decision-making, which workflow systems currently lack. Furthermore the major challenges created by high levels of uncertainty that are typical of clinical processes may suggest ways in which advanced decision models could be integrated into business process models. The task modelling approach may also offer benefits in terms of its “naturalness” for the end user’s perspective on a clinical process. Furthermore techniques from AI and knowledge engineering that are playing an important role in CIG development may also offer ways in which business process models may be enhanced, including temporal abstractions [21], and ontological modelling of tasks and goals [11].

3. Discussion and Future Directions

In this discussion we have briefly described a range of approaches to modeling clinical processes, outlining early techniques like PERT and GANTT charting and Petri nets, together with a more detailed summary of rule-based systems, business process and workflow models, and techniques for designing and enacting task networks. We now

[6] http://is.tm.tue.nl/research/patterns/products.htm

[7] http://www.workflowpatterns.com/yawl/index.php

[8] http://is.tm.tue.nl/research/patterns/products.htm

[9] http://www.workflowpatterns.com/yawl/index.php

draw some general, comparative conclusions and identify what we believe to be a major challenge for the development of computer-supported clinical services.

The earliest approaches to developing static models of processes continue to be influential. They still have important uses, in process planning for example, but do not offer process enactment or simulation capabilities. Petri nets were also originally intended as a formalism for static system models but have been increasingly successful in dynamic simulations and similar uses. The growth of interest of the workflow community in clinical process modelling and workflow management suggests that PN technology is an important candidate for designing and implementing clinical services in the future. However PNs do not provide the equivalent of modern programming languages or executable specification languages. Recent standardisation proposals for modelling systems based on Petri nets (e.g. Petri net Markup Language) and the development of enactment engines for PN applications, such as YAWL, may significantly change this position.

Over the last 30 years or so we have also seen the emergence and maturing of rule-based and logic programming technologies in the knowledge engineering and AI communities. Although these techniques were not developed with process management as a primary use-case they have always been associated with the development of practical applications such as medical expert systems. This has led to many proposals for languages for specifying knowledge-based services and development of specialised interpreters for those languages. As with Petri nets, logic based systems offer a powerful and sound foundation for designing and building clinical services.

On the face-of it rule-based systems are very different from Petri nets but they have some notable features in common. Transitions in Petri nets, condition ⇒ action rules in production systems, and premise ├ conclusion schemas in logic can all be viewed as concrete instances of a general *pre-condition* | *post-condition* signature.

In their simplest form technologies based on these abstract computational models share the dubious distinction of having been criticised for poor efficiency and ease of understanding. However analogous solutions to these problems have been found in all cases. Optimisation techniques have greatly mitigated efficiency problems in PN compilers, production rule systems and Prolog interpreters. The use of higher level objects on top of the abstract *pre-condition* | *post-condition* lead to greater naturalness and understanding for concrete application domains. The high level objects offered by workflow languages (pools, lanes, flow objects and messages) and task classes supported by TNLs (plans, actions and decisions) make the underlying formalisms much easier for non-technical users like clinicians to understand [9].

A question that would naturally seem to arise is whether PNs or logic, or some other kind of semantics, is best for formalising processes. We think this is not in fact a particularly useful question.

Logic has traditionally been concerned with defining what may be *validly or rationally inferred* from a set of facts and rules. Given data about a concrete clinical situation we typically wish to deduce valid conclusions from the data (e.g. the diagnosis or level of clinical risk). Classical propositional and predicate calculus provide the traditional logic frameworks, though various interpretations of probability theory have often been the preferred frameworks for dealing with uncertainty. In recent years computer science and cognitive science have also developed *non-classical logics*, including logics of time, space and causality for reasoning about the everyday physical world and *non-monotonic* logics and argumentation systems which extend non-classical logic into the domains of uncertainty, evidence-based reasoning and planning.

These and other logics are often held to capture patterns of “common-sense” human thinking. This combination of common sense reasoning and sound semantics should benefit the design of technologies for decision-making, care planning and process management - being at once demonstrably useful in clinical practice and intuitive for the doctors and patients who use them.

Petri nets, on the other hand, are neutral with respect to the nature of thinking about medicine or anything else, and have been developed only with process engineering objectives in mind. While task network models might make some claim to being grounded in forms of reasoning, decision-making and planning which resemble their counterparts in medical thinking, no comparable claim has been made to our knowledge that the concepts of workflows, pools, lanes etc capture natural “business thinking”.

It seems to us, however, that these two key approaches to modelling processes should not be seen as competitors. Rather they represent complementary views in which logic-based semantics provide an intuitive foundation for formalising medical knowledge and clinical thinking, while Petri nets provide a sound technical foundation for modelling organisations and dealing with problems like concurrent and distributed computation. Complex clinical organisations are highly distributed and exemplify many different forms of concurrency including multiple interrelated workflows, high bandwidth event and data transmission, asynchronous messaging, etc. In our opinion a unified approach to modelling complex clinical organisations and processes which builds on both theoretical traditions is a desirable research objective.

We close this discussion with some remarks on a future scenario for the delivery of complex, distributed and concurrent clinical services. This is the scenario of “careflow” mentioned earlier, which we believe challenges current capabilities, whatever our technical persuasion. Stefanelli[10] motivates the careflow concept with the observation that “… the problem of guideline dissemination and implementation in health care organisations needs to [consider that the] individual doctor-patient relationship is being replaced by one in which the patient is managed by a team of health care professionals, each specializing in separate aspects of the care process. This 'shared care' depends critically on the ability to share patient-specific information and medical knowledge easily among care providers. Indeed, it is the present inability to share clinical practice guidelines across systems and organizations that represents one of the major impediments to progress towards effective evidence-based care. Strategically, there is a need to take a more clinical process view of health care delivery and to identify the appropriate organizational and information infrastructures to support the process." The concept that Stefanelli believes is needed to implement a careflow service is that of the software agent.

A software agent is typically viewed as an autonomous entity that can take responsibility for providing services such as situation and risk assessment, decision-making and task management under significant levels of uncertainty, responding adaptively to changing circumstances and events [9]. In multi-agent systems agents with specialised capabilities interact to request specific information or services and/or pass responsibilities for achieving business or clinical goals to other specialist agents, and enter into “contracts” with other agents to carry out specific tasks or business processes. There is now a rapidly growing literature on multi-agent systems in healthcare applications (see http://www.openclinical.org/agents.html) though as yet

[10] http://www.openclinical.org/briefingpaperStefanelli.html

there are many more successful examples of individual guideline based services than multi-agent systems in clinical applications.

4. Research Agenda

Current approaches to implementing clinical decision support and guideline services need to advance significantly if we are to provide realistic support for complex clinical processes like extended treatment pathways and multidisciplinary shared care. Decision support technologies, task network models, workflow management systems and multi-agent technologies can all make significant contributions, but they appear to be individually inadequate for implementing integrated careflow services. All of these traditions have overlapping concerns, but often adopt very different languages and concepts, and have very different theoretical foundations (e.g. Petri nets and mathematical logic). In our view a primary research goal must be, so far as is possible, to unify the various techniques and resist the tendency towards a competitive fragmentation of the field.

We have concentrated on task network and workflow models in this discussion, but older traditions remain relevant and newer techniques including autonomous agents and knowledge representation methods from AI and the semantic web etc are likely to make contributions [9]. An early benefit of a research programme aimed at integration of the different approaches would be the translation of productive ideas from one field to another. For example, the extensive work on control patterns in workflow is already contributing to a deeper understanding of task network models ([17]. On the other hand techniques which are used in the clinical guidelines field, such as reasoning and decision-making under uncertainty, automated care planning could address the need for achieving greater flexibility in process management that is an increasing concern in the workflow community (e.g. [1]).

Flexibility is a pivotal research problem for all the research traditions. Few but the most bureaucratic human organisations and processes can be reduced to rigid procedures which, once laid down, must be followed without exception. In medicine, for example, a substantial proportion of patients have highly individual personal needs and clinical circumstances, respond to treatments in different ways, or not at all. Simply following the normal clinical pathway is not always desirable or even acceptable, and a technology that rigidly imposes such routines will be rejected. A critical research challenge is to be able to support the clinician in a flexible and adaptive way with respect to clinical goals not just standard procedures: “doing the right thing”, not just “doing the thing right”.

References

[1] Adams M, ter Hofstede A H M, Edmond D, and van der Aalst, W, Facilitating Flexibility and Dynamic Exception Handling in Workflows through worklets (2005) http://sky.fit.qut.edu.au/~adamsmj/05Worklets_CaiseForum.pdf

[2] Benson T, Page R “Electronic Care Pathways: Definition of Terms” NHS National Programme for IT (discussion paper) Version 1.2, 14 June 2004

[3] Bloom BS, de Pouvourville N, Chatre S, Jayadevappa R, Weinberg D (2004) Breast cancer treatment in clinical practice compared to best evidence and practice guidelines. *Br J Cancer* 90 (1): 26-30.

[4] Das S, Fox J, Elsdon D and Hammond P "A flexible architecture for autonomous agents" *Journal of Theoretical and Experimental Artificial Intelligence*, 9 (4), 407-440, 1997

[5] de Clercq PA, Blom JA, Korsten HH, Hasman A. "Approaches for creating computer-interpretable guidelines that facilitate decision support." *Artif Intell. Med.* 2004 May; 31 (1), 1-27.

[6] Field MJ, Lohr KN (Eds). Clinical Practice Guidelines: Directions for a New Program, Institute of Medicine, Washington, DC: National Academy Press, 1990.

[7] Fox J, Barber D, Bardhan K D "Alternatives to Bayes: a quantitative comparison with rule-based diagnostic inference" Meth. Inf. Medicne, 1980, 19 (4): 210-215

[8] Fox J, Johns N, Lyons C, Rahmanzadeh A, Thomson R, Wilson P (1997) PRO*forma*: a general technology for clinical decision support systems. *Comput Methods Programs Biomed* 54: 59-67

[9] Fox J and Das S *Safe and Sound: Artificial Intelligence in Hazardous Applications*, Cambridge: MIT Press, 2000.

[10] Fox J, Patkar V, Thomas R (2006) Decision support for healthcare: the PRO*forma* evidence base. *Informatics in Primary Care* 14(1): 49-54.

[11] Fox J, Alabassi A, Patkar V, Rose A, Black E, "An ontological approach to modelling tasks and goals" *Computers in Biology and Medicine*, Vol. 36, No. 7-8, pp. 837-856, 2006

[12] Freemantle N, Harvey E, Grimshaw JM et al (1996) The effectiveness of printed educational materials in changing the behaviour of health care professionals. Cochrane Collaboration. Cochrane Library. Issue 3. Oxford: Update Software.

[13] Garg AX, Adhikari NK, McDonald H, Rosas-Arellano MP, Devereaux PJ, Beyene J, Sam J, Haynes RB (2005) Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *JAMA* 293 (10): 1223-1238.

[14] Glasspool DW, Oettinger A, Smith-Spark J H, Castillo F C, Monaghan V, Fox J. "Supporting Medical Planning by Mitigating Cognitive Load". *Methods of Information in Medicine*, 46 (6): 636-640.

[15] Grol R, Grimshaw J (2003) From best evidence to best practice: effective implementation of change. *Lancet* 362: 1225 -1230

[16] Kawamoto K, Houlihan C A, Balas E A, Lobach D F "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success" *British Medical Journal* 2005; 330: p 765 (2 April).

[17] Mulyar N, van der Aalst W, Peleg M "A Pattern-based Analysis of Clinical Computer-Interpretable Guideline Modeling Languages" *Journal of the American Medical Informatics Association* 2007;14(6):781-787

[18] Panzarasa et al, 2004 "A careflow management system for chronic patients" Panzarasa S, Bellazzi R, Larizzab C, Stefanelli M *Proceedings Medinfo 2004.*

[19] Patkar V and Fox J "Clinical guidelines and care pathways: a case study applying PRO*forma* decision support technology to the breast cancer care pathway" (this volume).

[20] Peleg M, Tu S, Bury J, Ciccarese P, Fox J, Greenes RA, Hall R, Johnson PD, Jones N, Kumar A, Miksch S, Quaglini S, Seyfang A, Shortliffe EH, Stefanelli M. Comparing computer-interpretable guideline models: a case-study approach. *J Am Med Inform Assoc*. 2003 Jan-Feb; 10(1):52-68.

[21] Shahar, Y., Miksch, S., and Johnson, P. The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine* (14): 29-51, 1998 (see also www.openclinical.org/gmm_asbru.html).

[22] Shortliffe, E.H. *Computer-Based Medical Consultations: MYCIN*, Elsevier/North Holland, New York, 1976.

[23] Sutton D and Fox J "The Syntax and Semantics of the PROforma Guideline Modeling Language" *J. Am. Med. Inform. Assoc*. 2003; 10(5):433-443.

[24] van Melle W, Shortliffe E H, Buchanan B, "MYCIN: A Knowledge Engineer's Tool for Constructing Rule-Based Expert Systems" in Buchanan and Shortliffe (eds.): *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison Wesley (1984)

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-63

Chapter 4
Formal Methods for Verification of Clinical Practice Guidelines

Arjen HOMMERSOM [a,1], Perry GROOT [a], Michael BALSER [b], and Peter LUCAS [a]
[a] *Radboud University Nijmegen, The Netherlands*
[b] *Universität Augsburg, Germany*

Abstract. Formal methods play an important role in the development of software and hardware systems. In recent years, there has been a growing interest to apply these methods in the area of medical guidelines and protocols. This paper summarises these efforts, compares the approaches and discusses the role of formal methods in this area.

Keywords. Verification, (Interactive) Theorem Proving, Model Checking

Introduction

Formal methods are commonly defined as mathematics-based techniques for the specification, development, and verification of software and hardware systems [49,69]. A specification of a system is a mathematical description that describes *what* the system should do, i.e., the requirements or properties that should hold for an implementation of a system, written as well-formed statements in a formal logic. Given the specification, formal verification is the act of proving or disproving the correctness of an implementation with respect to the specification in a mathematically rigorous way. Many of such techniques have been developed. However, we will focus this chapter on those techniques that have been applied in the context of clinical guidelines or protocols, which are for a large part based on logic.

Formal methods can be exploited in relationship to medical guidelines in several ways. One choice is to consider the guideline as the 'system' that is being developed. Then, verification involves checking whether this guideline adheres to certain correctness or quality criteria, which is the main topic of this chapter. However, guidelines could also be considered as the golden standard for other types of systems. For example, in developing local protocols, the requirements are often derived from the more general guideline, if available. Similarly, if we look at the actual clinical practice as a system, e.g., formalised in terms of an electronic patient record, then verification may involve comparing medical actions against recommendations given by a guideline. We will only briefly address such other possibilities as they are discussed more elaborately in a Chapter 7 of this book.

[1] Corresponding Author: Arjen Hommersom, Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, the Netherlands; E-mail: arjenh@cs.ru.nl.

This chapter is structured as follows. In Section 1, we review, from a formal perspective, some of the different specification languages that have been proposed for modelling guidelines. Questions regarding formal semantics and expressiveness are addressed. In Section 2, we address the different type of properties that have been investigated in literature. Next, in Section 3, we look at some actual verification studies and methodologies. In Section 4, we briefly discuss formal methods in relation to protocol development and compliance checking, as described above. Finally, in Section 5, we discuss the role of formal methods in its relation to medical guidelines and especially the impact it may have on medical practice.

1. Guideline Specification

Several methods have been developed to support the modelling of guidelines. Specialised guideline modelling languages have been developed of which some have a formal semantics. Other authors have proposed the use of general-purpose logical languages for the specification of guidelines as many languages that have been proposed in artificial intelligence have rich structures and are likely to be expressive enough for specification of aspects of guidelines. Furthermore, as they are often based on logic, they have a formal semantics. On the other hand, they lack some intuitive primitives elements of specialised guidelines modelling languages such as “decisions” or (medical) “actions”.

1.1. General-purpose Formal Languages

Description logic [3] is a family of well-known knowledge representation formalisms, which has been used for the development of a wide range of applications. An important advantage is that they constitute a decidable fragment of first-order logic, and thus provide a clear syntax and semantics and make automatic verification possible. In [53] it is proposed to model certain aspects of practice guidelines in this logic, i.e., taxonomic relations (*chest x-ray is a type of x-ray*), mereologic order (*identification is part of the interview*), and temporal order (*physical examination precedes chest x-ray*). Examples of the use of description logics in the clinical domain can be found in e.g., [54].

Logic-based formalisation of medical guidelines has also been suggested in the context of agent modelling. In [12], guidelines are considered a set of social integrity constraints, formalised using standard logic with additional operators **H** (indicating a ground fact), **E** (indicating an expectation), and **EN** (indicating a *negative* expectation). For example, the following:

$$\begin{array}{l}\mathbf{H}\,(enter(Patient, emergency\_ward), T_0) \rightarrow \\ \mathbf{E}\,(examine(Physician, Patient), T_1) \wedge T_1 > T_0\end{array}$$

denotes that in whenever a patient enters the emergency ward, then it is expected that a physician will examine the patient at some later time instance T_1. The underlying idea is to use these constraints in order to prevent agent behaviour that is not compliant with the guideline. This is particularly useful in hospitals where integrity constraints could be used for checking if the hospital staff is compliant with the guideline. The semantics of this language has been formalised as an abductive proof procedure [1].

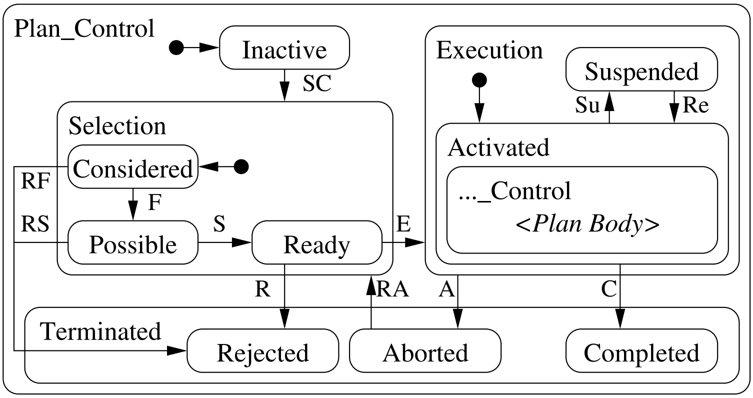


Figure 1. Plan state model of a single Asbru plan.

Temporal reasoning is an important aspect of medical guidelines. Therefore, it is not surprising that temporal logic has been proposed as a suitable formalism. For example, computation tree logic has been applied for modelling a guideline for the purpose of formally studying refinement of guidelines to protocols (cf. Chapter 7 and [30]).

1.2. Guideline Modelling Languages

A number of languages have been developed to write down clinical guidelines in a computer-interpretable format (so called computer-interpretable clinical guidelines, CIGs). In [38], the languages Asbru, EON, GLIF, GUIDE, PRODIGY, and PRO*forma* are compared, based on a common case study. The paper identifies a number of common components and a number of significant differences between languages.

1.2.1. Semantics

For the syntax and semantics of guideline languages three levels can be considered:

1. the dynamic behaviour of guideline components, and
2. conditions usually given in a so called expression language,
3. temporal abstraction of conditions.

For Asbru, the dynamic behaviour of plan execution (level 1) is defined in a formalism called Structural Operational Semantics (SOS) [44]. The abstract execution model of a single plan is illustrated in Figure 1. Each arrow in the state transition system represents a single SOS rule: Filter and setup conditions (F and S) are used to control the applicability of a plan, abort and complete conditions (A and C) are used to monitor execution. The sub plans in the plan body can be organised using different body types (e.g., sequential, any-order, or parallel). The current state of a plan – especially if a plan has been rejected, aborted, or completed – is propagated according to the plan hierarchy to its super and sub plans. If a plan is mandatory, it must be completed, otherwise it may also be rejected or aborted. Details can be found in [4,5]. Evaluation of conditions (level 2), however, is not considered in detail. Asbru offers a variety of possibilities to abstract from patient data (see [56,55,61]). In most cases, conditions are simple and can be translated one-to-one to a first order formula. For more complex conditions, a full formal semantics still needs to be defined. As a speciality of Asbru, conditions can be monitored over time according to so called time annotations (temporal abstraction, level 3). A revised and complete semantics of time annotations has been published in [52].

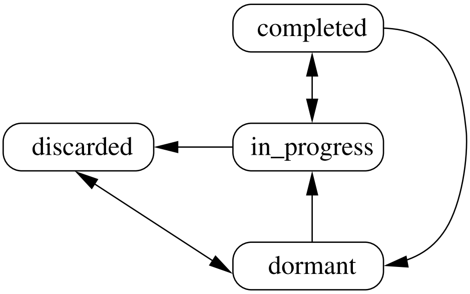


Figure 2. State and transitions of PRO*forma* Task component.

Similar to Asbru, an operational semantics for PRO*forma* is given in [63,24]. The dynamic behaviour of "Task" components is again described in an operational style. Figure 2 illustrates the underlying state transition system. The transition system has recently been simplified from 11 to 4 states without restricting expressiveness of the PRO*forma* language, as some of the states were found to be redundant to describe the core semantics of the language.

The semantics is described rather informally for GLIF [68], EON [65], GLARE [2, 25], and GUIDE [46]. The descriptions usually give a flavour of a semantics in terms of 'states' (e.g., completed and aborted states), message passing, etc, although these terms are not defined in detail. While this is enough for actually building guidelines, the semantics must be more precise when it comes to formal verification. Facilities for testing and making sure that the model is unambiguous and syntactically correct is available for most approaches. An overview of these facilities can be found in [16].

Verification, in particular a verification technique called model checking (cf. Section 3.3), is nevertheless applied to guidelines written in GLARE by translating GLARE clinical guidelines to the formal language Promela [25]. This translation can be considered as defining the formal semantics of GLARE. As a drawback, in order to fully understand the execution model of GLARE, one needs to fully understand the translation as well as the Promela semantics.

1.2.2. Temporal Constraints

Medical guidelines do not simply describe a process that is to be executed instantly. Instead, the process evolves in time where reasoning about time is done explicitly. As this is such a prominent feature of guideline representation languages, this topic deserves some more detail. As an example, consider the administration of a drug three times a day for five days in a row. Different guideline modelling languages offer different constructs for expressing temporal constraints. The temporal constraints normally define only certain minimum and maximum boundaries instead of fixed time periods; for example, a drug is not taken every day at exactly the same time but should be taken between 07:00 and 10:00 a.m.

In the following paragraphs, the constructs for expressing temporal constraints in Asbru and in GLARE are introduced.

Whereas temporal constraints in GLARE refer to the start and end point of actions, Asbru time annotations incorporate arbitrary conditions, e.g., conditions that refer to the condition of the patient. This is more expressive; however, it adds complexity to the verification problem.

Time Annotations in Asbru. An example property containing a temporal constraint is the following: "Intensive phototherapy should produce a decline in the TSB (total serum bilirubin) level of 1 to 2 mg/dl within 4 to 6 hours". This can be expressed with so called time annotations in Asbru:

$$\Delta_{\mathrm{TSB}} \leq -1\mathrm{mg/dl} \quad [\_, 6h]\ [4h, \_]\ [\_, \_] \quad \mathrm{enter}(\mathit{pti}, \mathrm{activated})$$

i.e., a condition Δ_{TSB} must occur 4 to 6 hours after the treatment plan 'pti' (phototherapy intensive) has been activated. In general, this is the pattern for Asbru time annotations:

$$\mathit{condition} \quad [\mathit{ESS}, \mathit{LSS}]\ [\mathit{EFS}, \mathit{LFS}]\ [\mathit{minDu}, \mathit{maxDu}] \quad \mathit{RP}$$

A starting interval is defined by an earliest starting shift (*ESS*) and a lasted starting shift (*LSS*). The earliest- and latest finishing shift (*EFS* and *LFS*) define the finishing interval. These intervals are relative to a reference point (*RP*), that describes a time point that is the offset to the shifts. In addition, a minimum and maximum duration (*minDu* and *maxDu*) can be defined. A complete formal semantics of Asbru time annotations is given in [52].

Temporal Constraints in GLARE. GLARE's temporal language allows one to model temporal constraints between actions, and is quite similar to Asbru's time annotations. In GLARE, it is possible to define the minimum and maximum *duration* of actions, the minimum and maximum *delay* between actions and repetition/periodicy constraints. Consider actions A and B. Then the following examples illustrate the types of relations that can be expressed:

- the duration of A is between 10 and 20 minutes.
- the end of A is equal to the start of B.
- B starts between 10 and 20 minutes after the end of A.
- A is repeated once every week for two weeks, until condition C does not hold anymore.

A more detailed description of time in the GLARE framework can be found in [2].

2. Specification of Guideline Properties

To be able to specify properties, a representation language is required. Similar to the previous section, general and specific languages have been proposed. First, we discuss the type of properties that have been investigated. Then, we focus on the general and more specific representation languages that have been used in literature. Finally, we focus some of the methodological issues related to the specification of properties.

2.1. Type of Properties

A wide range of properties of medical guidelines and protocols that have been checked can be found in literature. In order to make the distinction more comprehensible, it is useful to make a distinction between two classes of properties, namely:

1. *Intra-guideline consistency* (this term is proposed in [60]): this deals with consistency issues within a single guideline. Inconsistencies include structural defects, e.g., inconsistencies between temporal structure and temporal specifications; ambiguities, such as conflicting recommendations for the same patient group; incompleteness, when a recommendation is missing for a relevant patient group.
2. *Intentions / Goals*: these approaches deal with correctness criteria in terms of the process or result of executing the guideline. Note that we will use the term intentions and goals synonymously here.

The first type of properties are typically stated informally or in the language that the guideline is modelled in (e.g., logic). Hence, the focus on the representation of properties will be on the second type. These type of properties are derived from various sources. Shahar argues for explicitly specifying the design rationale in the guideline itself [57]. In practice, however, other sources have to be used, e.g., recommendations from other guidelines (inter-guideline consistency), indicators [66], expert opinions, and general criteria relative to additional medical knowledge.

2.2. Temporal Logic

2.2.1. Introduction

Several temporal logics have been developed, in particular tense logics since the 1960s. Differences between logics result from different models of time and expressiveness. In linear temporal logics (e.g., Linear Temporal Logic (LTL) [45]), models form a linear trace, while in branching logics models typically (e.g., Computation Tree Logic (CTL) [8,14,21]) form a tree.

A model of a medical guideline, is a Kripke structure M over a set of atomic propositions AP, which formally is defined as a four tuple $M = (S, S_0, R, L)$ where S is a finite set of states, $S_0 \subseteq S$ is the set of initial states, $R \subseteq S \times S$ is a total transition relation, and $L : S \to 2^{AP}$ is a function that labels each state with the set of atomic propositions true in that state. A *path* in the model M from a state s is an infinite sequence $\pi = s_0 s_1 s_2 \ldots$ such that $s_0 = s$ and $R(s_i, s_{i+1})$ holds for all $i \geq 0$. With π^i we denote the suffix of π starting at s_i, i.e., $\pi^i = s_i s_{i+1} s_{i+2} \ldots$.

CTL uses atomic propositions, propositional connectives, *path quantifiers* and *temporal operators* for describing properties of *computation trees*, i.e., the tree that is formed by designating a state in the Kripke structure as the initial state and then unwinding the structure into an infinite tree according to the transition relation R with the initial state as root. This leads to two types of formulas: *state formulas*, which are true in a specific state, and *path formulas*, which are true along a specific path. A path formula is build up by applying one of the temporal operators to state formulas. In this chapter, the temporal operators used are **X**, **G**, **F**, and **U**. With $\mathbf{X}\varphi$ being true if φ holds in the next state, $\mathbf{G}\varphi$ if φ holds in the current state and all future states, $\mathbf{F}\varphi$ if φ holds in the current state or some state in the future, and $\varphi\mathbf{U}\psi$ if φ holds until eventually ψ holds. A state formula can be

$$\begin{array}{lcl}
M, s \models p & \Leftrightarrow & p \in L(s) \\
M, s \models \neg f_1 & \Leftrightarrow & M, s \not\models f_1 \\
M, s \models f_1 \wedge f_2 & \Leftrightarrow & M, s \models f_1 \text{ and } M, s \models f_2 \\
M, s \models \mathbf{E} g_1 & \Leftrightarrow & \text{there is a path } \pi \text{ from } s \text{ such that } M, \pi \models g_1 \\
M, \pi \models f_1 & \Leftrightarrow & s \text{ is the first state of } \pi \text{ and } M, s \models f_1 \\
M, \pi \models \neg g_1 & \Leftrightarrow & M, \pi \not\models g_1 \\
M, \pi \models g_1 \wedge g_2 & \Leftrightarrow & M, \pi \models g_1 \text{ and } M, \pi \models g_2 \\
M, \pi \models \mathbf{X} g_1 & \Leftrightarrow & M, \pi^1 \models g_1 \\
M, \pi \models \mathbf{F} g_1 & \Leftrightarrow & \text{there exists a } k \geq 0 \text{ such that } M, \pi^k \models g_1 \\
M, \pi \models g_1 \mathbf{U} g_2 & \Leftrightarrow & \text{there exists a } k \geq 0 \text{ such that } M, \pi^k \models g_2 \text{ and} \\
 & & \text{for all } 0 \leq j < k, M, \pi^j \models g_1
\end{array}$$

Figure 3. Semantics of temporal logic with f_1 and f_2 representing state formulas and g_1 and g_2 representing path formulas.

built inductively from atomic propositions, propositional connectives, and if f and g are path formulas, then $\mathbf{E}f$ and $\mathbf{A}f$ are state formulas. The path quantifiers **A** and **E** are used to specify that all or some of the paths starting at a specific state have some property.

The semantics of CTL is defined with respect to a Kripke structure M. Given a state formula f, the notation $M, s \models f$ denotes that f holds in state s of the Kripke structure M. Assuming that f_1 and f_2 are state formulas and g_1 and g_2 are path formulas, the relation $\models$ is defined inductively as shown in Figure 3. The remaining syntax consisting of $\vee$, $\rightarrow$, **G**, **A** can be defined as usual, i.e., $f_1 \vee f_2 \equiv \neg(\neg f_1 \wedge \neg f_2)$, $f_1 \rightarrow f_2 \equiv \neg f_1 \vee f_2$, $\mathbf{G}g \equiv \neg \mathbf{F} \neg g$, and $\mathbf{A}f \equiv \neg \mathbf{E} \neg f$.

In contrast to CTL, LTL provides operators for describing events along a single computation path. Each formula is of the form $\mathbf{A}f$, with f being a path formula, which is either an atomic proposition or inductively defined as $\neg f$, $f \vee g$, $f \wedge g$, $\mathbf{X}f$, $\mathbf{F}f$, $\mathbf{G}f$, $f\mathbf{U}g$ with f, g path formulas. This language can be evaluated on Kripke structures as presented in Figure 3.

2.2.2. Expressiveness and Complexity

It is a well-known fact that the expressiveness of LTL and CTL is incomparable. For example, the following CTL statement

EF *normotension*

i.e., *'normotension' (normal blood pressure) may eventually occur* is not expressible in LTL. On the other hand, the following LTL formula

F G *normotension*

i.e., *eventually 'normotension' will always hold* is not expressible in CTL. Note that the formula **AF AG** *normotension* is stronger and expresses that there exists some state after which *all* patient groups have a normal blood pressure *at the same time*, which is more restrictive than the previous formula.

Both for theorem proving as well as model checking, reasoning using CTL is generally easier than reasoning in LTL, for example, LTL is in a higher complexity class. However, the discussion whether to use CTL or linear-time temporal logic (LTL) for model checking is far from being settled, as LTL is usually more intuitive and better suited as a specification language. For example, in [67], the advantages of linear-time frameworks is identified in terms of expressiveness, compositionality, property-specific abstractions, uniformity, and the use of bounded model checking.

2.3. Clinical Goal Representation

Several ontologies for intentions have been proposed in the context of medical guidelines.

2.3.1. Ontology of Goals in Breast Cancer

On the basis of a corpus of examples of clinical goal statements in breast cancer, an ontology was developed in [23]. As the authors describe, when clinical processes are designed and enacted, this should allow for the possibility of urgent changes to the care plan or "plan repair". In order to do this, a reason for each service has to be made explicit to be capable of recovering when goals are not achieved. They make a distinction between knowledge and action goals, where knowledge goals deal with acquiring information and deciding between alternative hypothesis about the world and action goals deal with achieving some state of the world and enacting tasks. A conclusion of this work is that this delivers a more balanced classification of types of goals compared to other ontologies. However, further work is expected to be needed in order to make a final scheme in the context of breast cancer.

2.3.2. Asbru Intentions

Although most guideline representation languages allow for representation of goals and intentions, representation of intentions is most developed in the Asbru language [38]. These intentions are considered "temporal patterns of provider actions and patient states, at different levels of abstraction, that should be maintained, achieved, or avoided" [57]. Furthermore, a distinction is made to whether the intention refers to a clinical state or action and whether the intention holds during enactment of the clinical process (*intermediate*) or after it has been completed (*overall*). The intention may also contain a rich temporal structure.

In the verification methods using Asbru, these intentions are typically formalised in temporal logic. For example, in [7], the Asbru intention "achieve overall state: α" is formalised as:

$$\mathbf{AG}\,(current\_plan = completed \rightarrow \mathbf{AF}\,\mathbf{AG}\,\alpha)$$

From a formal point this raises some questions. For example, in this example, it is possible, due to the use of the $\mathbf{F}$ operator, that α holds much later than the current plan. Moreover, one could argue that the similar looking, but non-equivalent, LTL formula:

$$\mathbf{G}\,(current\_plan = completed \rightarrow \mathbf{F}\,\mathbf{G}\,\alpha)$$

is the right formalisation. The point that we would like to make here is that formal languages are particularly useful to discuss such subtle differences. Even though in the original Asbru specification, the property seems uncomplicated, questions can be raised with respect to the intended semantics.

2.4. Methodology for the Specification of Properties

A major problem in the specification of properties as we have presented so-far is that properties from sources other than the original guideline differ in terminology. This is especially common in medicine where terms typically have multiple synonyms. Furthermore, properties may address aspects which are not even contained in the guideline. These properties can only be verified if the guideline is enriched by the additional aspects.

The problem of attaching the terminology of guidelines to the terminology found in properties is commonly recognised in the literature [26]. This problem could be further addressed using ontologies to standardise terminology as found in some of the guideline representation languages.

A structured approach to bridge the gap between informal properties and a temporal formula matching the aspects and terminology of the guideline has been proposed in [62]. This paper introduces a stepwise approach to formalise properties. The original informal goal is *reduced* in a first step to the scope of the guideline. In a second step, the goal is *normalised* to determine the expected behaviour and the timing constraints, i.e., start and end points between which the behaviour should be observed. After the normalisation step, it is rather easy to correctly *formalise* the property in a formalism called Goal Definition Language (GDL). A final step, the *attachment*, maps concepts and terminology of the property definition to concepts of the guideline.

This process does not solve the problem of mapping the different terminology of properties and guidelines. However, it gives structure to the process of attaching properties to guidelines and makes sure that medical domain experts are able to perform and understand the different steps. Assuming the domain experts are aware of both the ontology of the property as well as the ontology found in the guideline, this methodology enables the validation of properties.

3. Verification

Properties can be verified *on-the-fly*, i.e., properties define runtime constraints which are monitored during guideline execution, or *prior to execution*. Runtime constraints are always evaluated *for a given case* while verification prior to execution has to take into account *every possible case*. The latter is thus much more complex. This chapter is focussed on verification prior to execution.

There are roughly two verification approaches, namely model checking [13,47] which explores a (finite) model and theorem proving which explores logical derivations of a theory. There has been a particular focus on the use of theorem provers for reasoning about programs, which can be traced back the well-known Boyer-Moore theorem prover in the early 1960s [10,9]. In AI, theorem provers have for example been used to verify knowledge-based systems (e.g., [22]), as the knowledge representation is often based on logic.

3.1. Interactive Theorem Proving

The European project Protocure[2] has had a major impetus on the use of formal methods for the verification of medical guidelines. In [64], the results of this work is summarised. The guideline that is used deals with the treatment of jaundice and diabetes and is modelled in Asbru. This model was then, partly manually and partly mechanically, translated to temporal logic and given to an interactive theorem prover, called KIV[3] [6,27]. Indicators and intentions mentioned in the guideline were then used as correctness criteria in order to verify these guidelines. This work was subsequently extended in several ways. First, the semantics of a part of the Asbru language was incorporated in the KIV theorem prover, making it possible to translate a guideline model completely automatically [51], including complex time annotations that can occur in guidelines (for more details see Section 1.2.2). Second, the addition of background knowledge in order to verify more general quality criteria was investigated [29]. A complete description of the latter work can be found in [50].

Interactive theorem proving systems are sometimes called “proof assistants” as they do not construct proofs themselves, but rather support the construction of a proof by a user. In mathematics, proof assistants such as Mizar, HOL, and Coq are popular; in these systems almost all proof steps have to be performed manually. KIV was designed for use in program verification and attempts at providing more proof steps automatically; however, it does not exhaustively investigate large search spaces which keeps the amount of time it spends on calculations under control. The main advantage of such techniques is that it can, in principle, handle problems of arbitrary complexity, hence it is especially suitable if the model of the guideline is detailed and contains many complex constructs. By abstracting parts of the guideline, more automated techniques, such as automated theorem proving or model checking become feasible. These have also been applied to medical guidelines and are discussed below.

3.2. Resolution-based Theorem Proving

It was shown that for reasoning about models of medical knowledge, for example in the context of medical expert systems [33], classical automated reasoning techniques (e.g., [48,70]) are a practical option. In [31], the use of automatic theorem proving techniques for quality checking medical guidelines was studied. Translation of temporal logic yields a restricted first-order theory. Such a formalisation is suitable for use in standard resolution-based theorem provers. Typically, automated theorem provers require little or no interactions compared to interactive theorem provers. Resolution-based theorem proving facilities have been proven successful for many complex problems in algebra [43] and logic [32]; however, it does put a certain limit to the complexity of the guideline that can be verified.

3.3. Model Checking

With model checking [15], temporal properties can be automatically verified for a given state transition system. In principle, model checking is automatic, however, the applica-

[2] `http://www.protocure.org` [accessed February 2008]

[3] `http://www.informatik.uni-augsburg.de/swt/kiv` [accessed February 2008]

tion is limited to finite state transition systems. During the last years, tools have been refined, additional methods to automatically reduce the state space have been introduced, and computers, in general, have become more and more powerful such that nowadays, systems with a very large number of states can be automatically verified. Popular model checkers are SMV[4] [15], which uses Binary Decision Diagrams [35], SPIN[5] [28], an explicit state model checker, and others.

Model checking has become very helpful in software engineering for analysing reactive system designs. A medical guideline can be viewed as a concurrent system and model checking can be applied. It is necessary to transform the medical guideline to the input language of the model checker. Transformation can be automated by writing a suitable compiler. After the guideline has been transformed, temporal properties expressed either in CTL or in LTL can be verified. An interesting aspect of model checking is that, if the property does not hold, a counter example is provided which helps in improving the medical guideline or property. However, if verification does not terminate, the guideline model must be abstracted to reduce the state space. This abstraction must be provided manually such that, in general, model checking still requires expert knowledge.

In [7], the Cadence SMV model checker has been used to verify temporal properties of Asbru medical guidelines. An Asbru guideline is automatically translated to the SMV input language. A large subset of the Asbru language is translated, however, complex conditions and certain details of temporal constraints are currently abstracted. As a consequence, only a restricted set of properties can be verified. In the paper a selection of structural properties is considered. Verification of these properties has revealed a number of errors in the Asbru model.

GLARE medical guidelines have also been verified with model checking [25]. A guideline is translated to the input language of the SPIN model checker. This translation is fully automatic.

3.4. Other Techniques

Several other techniques have been proposed primarily for checking that the guideline model is internally consistent. They may be used as means to validation (i.e., check that the model represents the guideline) or verification (i.e., to check that the guideline is correct).

Rule-based and Decision Table. In [59,58], guidelines are represented as a decision table and completeness and ambiguousness are investigated of the guideline. In [36], guidelines are looked upon as rules, similar to modelling as done in for example *Arden Syntax*. Verification involves checking that the guideline is complete, i.e., for every possible situation an advice is given using a tool called "Commander". In their study of a guideline for childhood immunisation, they were able to identify a number of missing immunisation rules.

Coherence Analysis. In [18,20], Asbru models are translated to first-order logic and rich structural properties are being investigated in order to check the coherence of the model. Similar to the structure, the coherence of temporal constraints that have been put on the (Asbru) model is discussed in [19]. If problems with coherence can be traced back

[4] `http://www.cis.ksu.edu/santos/smv-doc` [accessed February 2008]

[5] `http://www.spinroot.com/` [accessed February 2008]

to the guideline, this can be considered a form of verification; however, the verification mostly deals with the formal model rather than checking the correctness of the original guideline.

Petri Nets. A different method to simulate dynamic systems is by modelling the system as a so called Petri net [41,42], which is also known as a place/transition net or P/T net. A Petri net is a mathematical representation of discrete distributed systems that graphically depicts the structure of a distributed system as a directed bipartite graph with annotations using place nodes, transition nodes, and directed arcs. A whole range of tools is available for using Petri nets to model and simulate complex dynamic systems.[6] Recently, this technique has been applied to analyse biological systems [40,37]. In [40], Petri nets have been used to model malaria parasites invading host erythrocytes. Petri nets also form the basis for the GUIDE guideline representation language, where they have been used for simulation of the health care processes [46].

3.5. Verification of Temporal Constraints

Checking temporal constraints is not impossible in a theorem proving and model checking approach. In fact, reasoning using temporal constraints been done using interactive theorem proving [51]. However, in many other cases other techniques are employed. One approach to deal with temporal constraints involves monitoring temporal constraints *during executing* of guidelines. This is possible in various guideline modelling frameworks. A more challenging approach is to verify the constraints *prior to execution*.

The temporal constraints in GLARE can be mapped to STP-trees , an exension of the “standard” STP (Simple Temporal Problems, see [17]) to cope with (possibly periodical) repeated actions. Algorithms exist to automatically verify the *consistency* of a given set of constraints prior to the execution of guidelines [2].

4. Applications of Formal Methods

4.1. Verification

Figure 4 shows a range of formal methods ranging from cheap and incomplete to very expensive and complete (loosely based on a picture by Rushby [49]). Many of the techniques discussed in Section 3.4 can be considered implicit formal methods (e.g., type checkers, parsers), as they are largely automatic and could be, for example, integrated in guideline authoring software. As the picture suggests, some other techniques, such as model checking and theorem proving require significant amount of work and is unlikely to be done by guideline developers. Nonetheless, as we have shown in this chapter, promising results have been reported in literature. Given the fact that guidelines have a large impact in medical practice, costs associated with the use of formal methods might be justified. We elaborate on this point somewhat further in the final section.

[6] `http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html` [accessed February 2008]

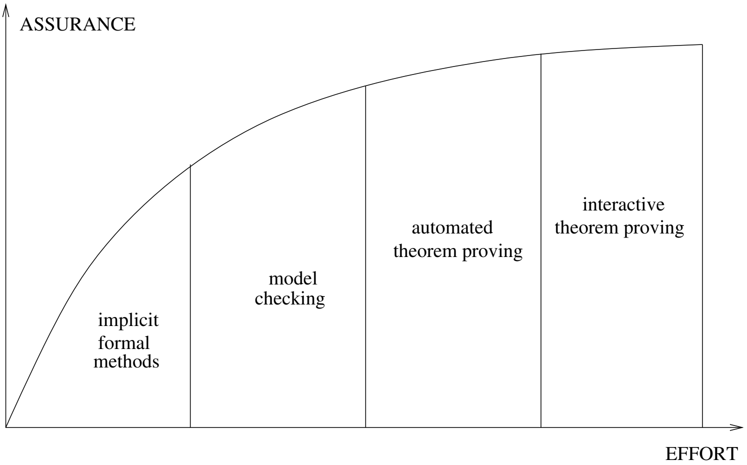


Figure 4. A spectrum of formal methods for formal verification allowing a tradeoff in the properties one can verify (assurance dimension) against the effort one needs to invest to obtain results (effort dimension).

4.2. Adaptation

Guideline adaptation is a process in which existing guidelines are adapted so that they can be used within a different care setting. Several reasons may exist for adapting the recommendations of an existing guideline to suit a local context, e.g., cultural differences, constraints on resources, end-user involvement, etc, though it is often the case, that the adaptation is faithful to the original guideline [34,30,39]. It is possible to employ formal methods in order to make sure that the adaptations do not violate the original guideline. A review of adaptations as done in practice can be found in Chapter 7 of this book.

4.3. Compliance

More general to the idea of using formal methods for checking that an adaptation is compliant to the guideline is checking that the actual clinical process is compliant to the guideline. If the clinical process is recorded as in, for example, an electronic patient record, then verification can be done on this basis. The use of formal methods can then be described as *critiquing*, i.e., to find differences between the actual actions and a set of 'ideal' actions as described by a clinical guideline. This topic is further discussed in Chapter 9 of this book.

5. Discussion

In this chapter, we have given an overview on the use of formal methods in the analysis of medical guidelines. In guideline specification, we found that there are quite a number of specification languages that have been proposed, but only a limited amount of work has been done in providing a formal semantics of these languages, which renders only a few of them suitable for the use of formal methods. Research in the area of specification of properties has mainly focused on (1) the acquisition of properties, and (2) design of an ontology of properties. Languages that are used are either informal, which again makes it problematic to use them in formal methods, or are closely based on a standard temporal logic. Verification has moved forward the last few years from the ad-hoc use of formal models in order to analyse certain aspects of guidelines to more systematic approaches using techniques that are now widely used in the formal methods community such as theorem proving and model checking.

In medicine, safety is extremely important, witnessed by the fact that ensuring safety is the primary preoccupation of regulatory agencies. Nonetheless, mistakes are made in hospitals; in fact, it was found that every year, in the Netherlands, 30,000 people are harmed and 1,700 people die in hospitals due to causes that can be avoided [11]. As medical errors have such far reaching consequences, there is good reason to investigate in the use of formal methods in medicine. Guidelines play an important role and errors in these guidelines may contribute to medical errors and mistakes. It is therefore clear that making sure that the guidelines are of the highest possible quality is essential.

We believe the benefits of using formal methods on top of other techniques to improve guidelines speak for themselves. First, formal verification and especially interactive verification is very helpful in analysing the language itself as formal methods force one to formalise the semantics. For example, this resulted in a number of problems with the Asbru language, which were detected while verifying properties of a medical guideline. As a consequence, the formal semantics of time annotations in Asbru has been significantly improved by verifying properties of the language itself [52]. The same holds for medical guidelines itself: much can be gained by formalising medical guidelines in practice. Informal text is interpreted differently by different readers and it is difficult to keep all parts of an informal medical guideline consistent, as a guideline is typically written by various authors. However, the true challenge remains to introduce a standardised formal language into the practice of guideline development. Only after this challenge has been met, the true potential of formal verification can be seen.

6. Research Agenda

As mentioned in the discussion, significant progress has been made in the last few years in the area of formal methods and clinical guidelines, just considering the amount of work that has been produced. However, much work still has to be done. Some of the issues that could be further investigated are mentioned here.

First, it would be convenient to introduce a standardised, machine readable format into the guideline development process. Otherwise a gap remains between informal text of the medical guideline and the machine readable model which is the basis for further analysis. The machine readable format must be easy to understand and yet expressive enough for a large variety of medical guidelines. Currently, a number of standardised languages exist for writing down medical guidelines, yet none of these languages have been used by guideline developers on a large scale, nor do they take into account the special features of formal verification. Similarly, a detailed formal semantics should underly the machine readable format and should be used as a standard for building tools such as editor, interpreters, compilers, etc.

Formal verification of properties is difficult and time consuming. While interactive verification can only be performed by logicians, automatic methods have potential to be applied by guideline designers. This raises the question in which situation a certain technique should be employed. Guidelines for guideline developers could improve the practical usefulnes as well as the visibility of the research that is being done.

Finally, and what is possibly most challenging is that there seems to be a gap between the work in this area that has been done so far and the medical community. It seems to be notoriously difficult to get medically relevant results, which might be due to the fact that only very few medical doctors are involved in this research. For example, it is relevant to know whether or not a guideline is “safe” or “correct”, which are concepts that are difficult to grasp. However, in order to make a real impact in medicine, such difficult questions will have to be answered.

References

[1] M. Alberti, M. Gavanelli, E. Lamma, P. Mello, P. Torroni, and G. Sartor. Mapping deontic operators to abductive expectations. *Computational & Mathematical Organization Theory*, 12(2):205–225, 2006.

[2] L. Anselma, P. Terenziani, S. Montani, and A. Bottrighi. Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. *Artificial Intelligence Medicine*, 38(2):171–195, 2006.

[3] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider. *The Description Logic Handbook - Theory, Implementation and Applications*. Cambridge University Press, 2003.

[4] M. Balser, C. Duelli, and W. Reif. Formal semantics of Asbru - an overview. In H. Ehrig, B. Kraemer, and A. Ertas, editors, *Proceedings of the 6th Biennial world Conference on Integrated Desgin and Process Technology*, pages 1–8, USA, 2002. Society for Desgin and Process Science.

[5] M. Balser, C. Duelli, W. Reif, and J. Schmitt. Formal semantics of Asbru – v2.12. Technical report, University of Augsburg, June 2006.

[6] M. Balser, W. Reif, G. Schellhorn, K. Stenzel, and A. Thums. Formal system development with KIV. In T. Maibaum, editor, *Fundamental Approaches to Software Engineering*, number 1783 in LNCS, pages 363–366. Springer-Verlag, 2000.

[7] S. Bäumler, M. Balser, A. Dunets, W. Reif, and J. Schmitt. Verification of medical guidelines by model checking – a case study. In A. Valmari, editor, *Proceedings of 13th International SPIN Workshop on Model Checking of Software*, volume 3925 of *LNCS*, pages 219–233. Springer-Verlag, 2006.

[8] M. Ben-Ari, Z. Manna, and A. Pnueli. The temporal logic of branching time. *Acta Inform*, 20, 1983.

[9] R. S. Boyer, M. Kaufmann, and J. S. Moore. The Boyer-Moore theorem prover and its interactive enhancement. *Computers and Mathematics with Applications*, 29(2):27–62, 1995.

[10] R. S. Boyer and J. S. Moore. Proving theorems about LISP functions. *Journal of the Association for Computing Machinery*, 22(1):129–144, 1975.

[11] M.C. de Bruijne, M. Zegers, L.H.F. Hoonhout, and C. Wagner. *Onbedoelde schade in Nederlandse ziekenhuizen: dossieronderzoek van ziekenhuisopnames in 2004.* Instituut voor Extramuraal Geneeskundig Onderzoek (NIVEL), 2007.

[12] F. Chesani, A. Ciampolini, P. Mello, M. Montali, P. Torroni, M. Alberti, and S. Storari. Protocol specification and verification by using computational logic. In F. Corradini, F. de Paoli, E. Merelli, and A. Omicini, editors, *WOA 2005: Dagli Oggetti agli Agenti. 6th AI*IA/TABOO Joint Workshop “From Objects to Agents”: Simulation and Formal Analysis of Complex Systems*, pages 184–192. Pitagora Editrice Bologna, November 2005.

[13] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite state concurrent systems using temporal logic. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[14] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Logic of Programs: Workshop*, number 131 in LNCS, Yorktown Heights, NY, May 1981. Springer.

[15] E.M. Clarke, O. Grumberg, and A.D. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, London, England, 2001.

[16] P.A. de Clercq, J.A. Blom, H.H.M. Korsten, and A. Hasman. Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial Intelligence in Medicine*, 31:1–27, 2004.

[17] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1–3):61–95, 1991.

[18] G. Duftschmid and S. Miksch. Knowledge-based verification of clinical guidelines by detection of anomalies. *OEGAI Journal*, pages 37–39, 1999.

[19] G. Duftschmid, S. Miksch, and W. Gall. Verification of temporal scheduling constraints in clinical practice guidelines. *Artificial Intelligence in Medicine*, 25:93–121, 2002.

[20] G. Duftschmid, S. Miksch, Y. Shahar, and P. Johnson. Multi-level verification of clinical protocols. In *Proceedings of the Workshop on Validation and Verification of Knowledge-Based Systems*, Trento, Italy, 1998.

[21] E.A. Emerson and E.M. Clarke. Characterizing correctness properties of parallel programs using fixpoints. In *Automata, Languages and Programming*, number 85 in LNCS, pages 169–181, 1980.

[22] D. Fensel, A. Schonegge, R. Groenboom, and B. Wielinga. Specification and verification of knowledge-based systems. In B.R. Gaines and M. A. Musen, editors, *Proceedings of the 10th Banff knowledge acquisition for knowledge-based systems workshop*, pages 1–20, Department of Computer Science, University of Calgary, 1996.

[23] J. Fox, A. Alabassi, E. Black, C. Hurt, and T. Rose. Modelling clinical goals: a corpus of examples and a tentative ontology. *Stud Health Technol Inform*, 101:31–45, 2004.

[24] J. Fox and S. Das. *Safe and Sound: Artificial Intelligence in Hazardous Applications*. AAAI Press, 2000.

[25] L. Giordano, P. Terenziani, A. Bottrighi, S. Montani, and L. Donzella. Model checking for clinical guidelines: an agent-based approach. In *AMIA 2006*, pages 171–195, Washington, 2006. American Medical Informatics Association.

[26] P. Groot, A. Hommersom, P. Lucas, R. Serban, A. ten Teije, and F. van Harmelen. The role of model checking in critiquing based on clinical guidelines. In R. Bellazzi, A. Abu-Hanna, and J. Hunter, editors, *11th Conference on Artificial Intelligence in Medicine*, number 4595 in LNAI, pages 411–420. Springer-Verlag Berlin Heidelberg, 2007.

[27] D. Haneberg, S. Bäumler, M. Balser, H. Grandy, F. Ortmeier, W. Reif, G. Schellhorn, J. Schmitt, and K. Stenzel. The user interface of the kiv verification system - a system description. *ENTCS special issue*, 2007. To appear.

[28] G.J. Holzmann. *The Spin Model Checker: Primer and Reference Manual*. Addison-Wesley Professional, 2004.

[29] A.J. Hommersom, P. Groot, P.J.F. Lucas, M. Balser, and J. Schmitt. Verification of medical guidelines using background knowledge in task networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(6):832–846, 2007.

[30] A.J. Hommersom, P. Groot, P.J.F. Lucas, M. Marcos, and B. Martínez-Salvador. *Computer-based medical guidelines and protocols: A primer and current trends*, chapter A Constraint-based Approach to Medical Guidelines and Protocols. IOS Press, 2008.

[31] A.J. Hommersom, P.J.F. Lucas, and P. van Bommel. Automated theorem proving for quality-checking medical guidelines. In *Proceedings of CADE-20 Workshop on Empirically Successful Classical Automated Reasoning (ESCAR)*, 2005.

[32] T. Jech. OTTER experiments in a system of combinatory logic. *Journal of Automated Reasoning*, 14(3):413–426, 1995.

[33] P.J.F. Lucas. The Representation of Medical Reasoning Models in Resolution-based Theorem Provers. *Artificial Intelligence in Medicine*, 5:395–419, 1993.

[34] M. Marcos, B. Martínez-Salvador, A.J. Hommersom, P. Groot, P.J.F. Lucas, A. Jovell, and S. Blancafort. Deliverable 5.1: Case-study in transformations for protocol development, 2006.

[35] K. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, Boston, 1993.

[36] D.W. Miller, S.J. Frawley, and P.L. Miller. Using semantic constraints to help verify the completeness of a computer-based clinical guideline for childhood immunization. *Computer Methods and Programs in Biomedicine*, 58(3):267–280, March 1999.

[37] M. Peleg, D. Rubin, and R.B. Altman. Using petri net tools to study properties and dynamics of biological systems. *J Am Med Inform Assoc.*, 2005.

[38] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E.H. Shortliffe, and M. Stefanelli. Comparing computer-interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association*, 10(1):52–68, 2003.

[39] M. Peleg, D. Wang, A. Fodor, S. Keren, and E. Karnieli. *Computer-based medical guidelines and protocols: A primer and current trends*, chapter Adaptation of practice guidelines for clinical decision support: a case study of diabetic foot care. IOS Press, 2008.

[40] M. Peleg, I. Yeh, and R.B. Altman. Modelling biological processes using workflow and petri net models. *Bioinformatics*, 18(6), 2002.

[41] J.L. Peterson. Petri nets. *ACM Computing Surveys*, 9(3):223–252, 1977.

[42] J.L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall, PTR Upper Saddle River, NJ, USA, 1981.

[43] J.D. Phillips and P. Vojtěchovskiý. Linear groupoids and the associated wreath products. *Journal of Symbolic Computation*, 40(3):1106–1125, 2005.

[44] G.D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Aarhus University, 1981.

[45] A. Pnuelli. A temporal logic of concurrent programs. *Theoretical Computer Science*, 13:45–60, 1981.

[46] S. Quaglini, M. Stefanelli, A. Cavallini, G. Micieli, C. Fassino, and C. Mossa. Guideline-based careflow systems. *Artificial Intelligence in Medicine*, 20(1):5–22, 2000.

[47] J-P. Queille and J. Sifakis. Specification and verification of concurrent systems is CESAR. In *International Symposium on Programming, LNCS 137*, pages 337–351. Springer Verlag, 1982.

[48] J.A. Robinson. Automated Deduction with Hyperresolution. *International Journal of Computatational Mathematics*, 1:23–41, 1965.

[49] J. Rushby. Formal methods and their role in the certification of critical systems. Technical report, CSL-95-1, March 1995.

[50] J. Schmitt, M. Balser, S. Bäumler, A. Dunets, N. Moebius, F. Lautenbacher, W. Reif, M. Marcos, Martínez-Salvador, A.J. Hommersom, P. Groot, P.J.F. Lucas, R. Serban, A. ten Teije, F. van Harmelen, A. Seyfang, S. Miksch, K. Rosenbrand, J. Wittenberg, and J. van Croonenborg. Deliverable d4.2c: Formal verification of selected guideline properties, 2006. Available at `http://www.protocure.org`.

[51] J. Schmitt, A. Hoffmann, M. Balser, W. Reif, and M. Marcos. Interactive verification of medical guidelines. In *14th Symposium on Formal Methods (FM'06)*, volume 4085 of *LNCS*, pages 21–27. Springer, 2006.

[52] J. Schmitt, W. Reif, A. Seyfang, and S. Miksch. Temporal dimension of medical guidelines: The semantics of asbru time annotations. In *ECAI 2006 WS – AI techniques in healthcare: evidence-baded guidelines and protocols*, 2006.

[53] S. Schulz and U. Hahn. A description logic approach to clinical guidelines and protocols. In *Symposium on Computerized Guidelines and Protocols*, 2004.

[54] S. Schulz, Markó. K., and B. Suntisrivaraporn. Complex occurrents in clinical terminologies and their representation in a formal language. In *Proc. of the First European Conference on SNOMED CT (SMCS'06)*, Copenhagen, Denmark, 2006.

[55] A. Seyfang and S. Miksch. Advanced temporal data abstraction for guideline execution. In *Proceedings of the Symposium on Computer-based Support for Clinical Guidelines and Protocols*, pages 88–102, 2004.

[56] A. Seyfang, S. Miksch, W. Horn, M. Urschitz, C. Popow, and C. Poets. Using time-oriented data abstraction methods to optimize oxygen supply for neonates. In *Artificial Intelligence in Medicine*, pages 217–226, 2001.

[57] Y. Shahar, S. Miksch, and P. Johnson. The asgaard project: A task-specific framework for the application and critiquing of time-orientied clinical guidelines. *Artificial Intelligence in Medicine*, 14:29–51, 1998.

[58] R.N. Shiffman. Representation of clinical practice guidelines in conventional and augmented decision tables. *Journal of the American Medical Informatics Association*, 4:382–393, 1997.

[59] R.N. Shiffman and R.A. Greenes. Improving clinical guidelines with logic and decision-table techniques: Application in hepatitis immunization recommendations. *Med Decis Making*, 14:245–254, 1994.

[60] O. Sokolsky, B.G. Silverman, and I. Lee. Consistency of clinical guidelines. Position Statement (`http://www.glif.org/workshop/position_stmt/ConsistencyPosition.pdf`), 2000.

[61] M. Stacey and C. McGregor. Temporal abstraction in intelligent clinical data analysis: A survey. *Artificial Intelligence in Medicine*, 39(1):1–24, 2007.

[62] R. Stegers, A. ten Teije, and F. van Harmelen. From natural language to formal proof goal: Structured goal formalisation applied to medical guidelines (extended abstract). In S. Staab and V. Svatek, editors, *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW'06)*, LNAI. Springer-Verlag, 2006.

[63] D. Sutton and J. Fox. The syntax and semantics of the PRO*forma* guideline modeling language. *Jounal of the American Medical Informatics Association*, 10(5):433–443, 2003.

[64] A. ten Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen, P. Lucas, S. Miksch, W. Reif, K. Rosenbrand, and S. Seyfang. Improving medical protocols by formal methods. *Artificial Intelligence in Medicine*, 36(3):193–209, 2006.

[65] S.W. Tu and M.A. Musen. From guideline modeling to guideline execution: Defining guideline-based decision-support services. In *Proc. AMIA Symposium*, pages 863–867, Los Angeles, CA, 2000.

[66] M. van Gendt, A. ten Teije, R. Serban, and F. van Harmelen. Formalising medical quality indicators to improve guidelines. In *Proceedings of the Tenth European Conference on Artificial Intelligence in Medicine (AIME'05)*, volume 3581 of *LNAI*. Springer Verlag, 2005.

[67] M.Y. Vardi. Branching vs. linear time: Final showdown. *Lecture Notes in Computer Science*, 2031:1–22, 2001.

[68] D. Wang, M. Peleg, S.W. Tu, A.A. Boxwala, O. Ogunyemi, Q. Zeng, R.A. Greenes, V.L. Patel, and E.H. Shortliffe. Design and implementation of the glif3 guideline execution engine. *Journal of Biomedical Informatics*, 37(5):305–318, October 2004.

[69] J.M. Wing. A specifier's introduction to formal methods. *IEEE Computer*, 23(9):8–24, 1990.

[70] L. Wos, R. Overbeek, E. Lusk, and J. Boyle. *Automated Reasoning: Introduction and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-81

Chapter 5
The Temporal Aspects of Clinical Guidelines

Paolo TERENZIANI[a,1], Efrat GERMAN[b], and Yuval SHAHAR[b]
[a]*Department of Computer Science, Universita' del Piemonte Orientale "A. Avogadro", Alessandria, Italy*
[b]*Medical Informatics Research Center, Department of Information Systems Engineering, Ben Gurion University, Beer-Sheva, Israel*

Abstract. Temporal aspects play a major role within clinical guidelines. Temporal issues arise when considering both guidelines per se, and the application of guidelines to specific patients. As a matter of fact, guidelines per se specify different diagnostic and\or therapeutic patterns, and temporal constraints on the intended times of execution of the actions they contain are an intrinsic part of guidelines themselves. Moreover, guidelines must be executed on the basis of patients' data, which are intrinsically temporal data (consider, e.g., the time when symptoms hold). Devising suitable *representation* formalisms to properly model such pieces of temporal information is a challenging task, for which several solutions have been proposed in the last years. Besides representation formalisms, *temporal reasoning* methodologies are also needed. *Temporal abstraction* is needed in order to infer abstract temporal data (as described in guideline action conditions) from "raw" timestamped patient data. Moreover, *temporal constraint propagation* is also needed, both at acquisition and at execution time. During acquisition, temporal constraint propagation is used to detect whether the temporal constraints in the guideline are consistent. At execution time, it is needed in order to check whether the actual time of execution of actions has respected the temporal constraints in the guideline, and to detect which are the next candidate actions to be executed, on the basis of the temporal constraints in the guideline. This chapter sketches some of the most important recent results about the above issues.

Keywords. temporal representation languages, temporal abstraction, temporal constraint propagation, timestamped data, qualitative and quantitative temporal constraints, temporal patterns, temporal databases

Introduction

The human way of perceiving and understanding the world deeply incorporates the notion of time. Therefore, temporal information is fundamental in most tasks, including the treatment of clinical guidelines. Specifically, the computer-based management of clinical guidelines involves many different issues, ranging from the formalization and

[1] Corresponding author: Paolo Terenziani, Universita' del Piemonte Orientale "Amedeo Avogadro", Via Bellini 25\g, 15100 Alessandria, Italy. Email: terenz@mfn.unipmn.it

acquisition of guideline content to the execution of an acquired guideline, interacting with the patients' data in the clinical database. Time plays a fundamental role in most of such issues. Specifically, in the guideline environment, temporal information play a fundamental role both as concerns the treatment of patients' data, and as concerns the definition of the temporal control flow of actions in a guideline.

Concerning patients' data (clinical records), it is worth stressing that most clinical data are naturally temporal. In order to be meaningfully interpreted, patients' symptoms, laboratory test results, and, in general, all clinical data, must be paired with the time in which they hold (called *valid time* henceforth). Additionally, for legal purposes, or for justifying physicians' decisions, also the time when such data have been inserted into the database might be stored (called *transaction time* henceforth). Some principled and general-purpose treatment of such temporal issues has been proposed within the research area of temporal databases, and some refinements and\or alternative proposals have been proposed within the medical informatics community.

Concerning clinical guidelines themselves, temporal information plays a fundamental role in at least two different ways. First, guidelines always contain temporal conditions on the patient data, e.g., to check the applicability of a given action. In several cases, such conditions define complex temporal patterns, that patients' data need to satisfy. Second, temporal constraints on the possible relative order and time of execution of actions is an intrinsic and important part of most clinical guidelines. For instance, in most therapies actions have to be performed according to a set of temporal constraints concerning their relative order, their duration, and the delays between them. Additionally, guideline actions must often be repeated at regular (i.e., periodic) times. Furthermore, it is also necessary to carefully take into account the (implicit) temporal constraints derived from the hierarchical decomposition of actions into their components and from the control-flow of actions in the guideline. Several general-purpose languages have been defined in the Artificial Intelligence community in order to model temporal constraints and temporal patterns. However, the peculiarity and richness of temporal constraints and patterns within clinical guidelines has demanded for new formalisms to represent temporal information.

Unfortunately, the effort needed to *represent* and *store* temporal data is almost worthless, if it is not paired with the development of suitable *temporal reasoning* techniques making such an information practically useful within the guideline context.

First of all, "raw" patient data stored in the clinical record (e.g., as obtained from a monitoring system) are usually too low-level to be useful within a guideline system, since temporal conditions in the guidelines usually demand for "high-level" complex temporal patters, e.g., as preconditions for action execution. For instance, trends or even more complex temporal patterns need to be abstracted from "raw" clinical data. Such a form of temporal reasoning, usually called *temporal abstraction*, is a demanding issue within the medical informatics context, and many different approaches have been devised in the last decade.

Second, temporal inference is needed on the temporal constraints on the possible relative order and time of execution of actions. Such a form of temporal reasoning is usually performed through the application of suitable *temporal constraint propagation* techniques. First of all, after (or during) a guideline acquisition, temporal constraint propagation is needed in order to check whether the temporal constraints in the guideline are consistent (i.e., whether there is at least a possible execution of the guideline that satisfies all the given constraints; otherwise, of course, the guideline is not executable!). Temporal constraint propagation is even more important at execution time, when a specific guideline is executed on a specific patient. As a matter of facts, it is an essential component of the execution engine, since it must be used in order to determine the next candidate actions to be executed, on the basis of the execution-time of the actions already executed, and of the temporal constraints in the guideline. A posteriori, temporal reasoning can be used in order to check whether actions have been executed according to the temporal constraints in the guideline.

In Artificial Intelligence and in Temporal Databases, several domain-independent approaches have been developed in order to represent and store temporal data, and to make inferences on them. However, the context of clinical guidelines presents several peculiarities, and this fact has motivated the extension of approaches already in the literature, as well as the development of new approaches. In particular, among the problems that make the treatment of temporal information in the clinical guideline context particularly challenging, we mention:

1. Temporal indeterminacy, i.e., the fact that the time of occurrence of clinical data may be only partially known, and that the sequence of actions in a guideline may be only partially specified
2. Temporal granularity, i.e., the fact that usually data may be defined at different levels of temporal precision (compare, e.g., monitoring vs diagnostic data)
3. Expressiveness, i.e., the fact that temporal constraints in the guidelines' conditions and\or specifying the control flow of execution may be quite complex, often including both qualitative and quantitative temporal constraints, and (possibly periodic) repetition patterns.

The chapter is organized as follows. In section 1, we deal with the representation of "raw" temporal patient data, sketching some of the most relevant results concerning temporal databases, and their impact on the treatment of clinical data. In section 2, we focus on the representation of temporal information in the guidelines. The inferential issues are coped with in sections 3 and 4. Specifically, in section 3 we overview the main approaches to clinical temporal abstraction, and in section 4 we sketch the main results about temporal constraint propagation, with specific focus on the results obtained within the area of clinical guidelines management. Finally, section 5 highlights some of the more interesting future developments in this area of research.

1. Storing Patients' Data: Temporal Databases

Clinical records containing patients' data are usually stored using (mostly relational) databases. However, patients' data are intrinsically *temporal* and, unfortunately, the research about temporal data has widely demonstrated that the simple addition of some timestamped attributes (e.g., the START and END times for the valid time of a tuple) is not enough, since many complex problems need to be tackled. Designing, querying and modifying time-varying tables requires a different set of techniques.

As an example, consider the *JOIN* between two temporal tables R_1 and R_2 (e.g., deriving from the execution of a "*FROM* R_1,R_2" clause in a standard SQL query). How can the time of the resulting table be obtained? "Standard" non-temporal join simply "link together" into a new tuple each pair of tuples $t_i \in R_1$, $t_j \in R_2$: each new tuple in the result is simply obtained as the set of values of the two tuples composing it. However, such a simple procedure cannot be applied in case tables R_1 and R_2 are temporal. As a matter of fact, it is not acceptable that the temporal tuples resulting from the *JOIN* have two different times (one from each composing tuple): a unique time should be provided for each resulting tuple, elaborated from the times of the tuples composing it. Within the temporal database community there is a shared agreement that *intersection* must be used in order to get the resulting time, but traditional SQL-based DBMS do not provide any hint or facility to do that, so that the solution of the problem is completely left to the programmer who is building the temporal application. Even more severe (and hidden) problems arise when coping with *PROJECTION* (e.g., deriving from the execution of a *"SELECT $attr_1,...attr_n$"* clause in a standard SQL query, whenever $attr_1,...attr_n$ is a proper subset of the attributes of the input tables), *union*, *difference* and *intersection* operations, *aggregate functions*, and so on.

In general, the problem is that temporal attributes do not behave like the other non-temporal attributes, since they model the time when the other non-temporal attributes hold. In other words, *time has a peculiar semantics*, and capturing it in a database is a challenging issue. In principle, it is possible to try to handle such issues in standard data models, generally at the expense of high data redundancy, awkward modeling, and complicated query languages.

But the real solution, as outlined by the last twenty years of research about (temporal) databases, is to provide once-and-forall a principled solution. This is the spirit that has animated more than 20 years of research within the temporal database (TDB henceforth) community. For the sake of brevity, we briefly sketch only some of the main results provided in this area (some general overviews are, e.g., [1,2]; a cumulative bibliography can be found in [3]). In particular, we end the section providing also a sketchy overview of some of the approaches developed in order to deal specifically with clinical temporal data.

A temporal database is a collection of time-referenced data. This can be either the sequence of versions (a transaction-time database), a history of the reality modeled by the database (a valid-time database), or both. Time impacts all aspects of a database, from its design (at the conceptual, logical, and physical levels), through its implementation, the query and modification languages, the data structure utilized, query optimization and query evaluation, and transaction processing.

A "data model" consists of two components, namely a set of objects and a language for querying those objects. Regarding the data model, *temporal conceptual models* generally extend the Entity-Relationship model [4] providing a formalism to elicit and represent temporal data semantics during conceptual design (see, e.g, the survey in [5]).

Temporal logical models generally extend the relational model. Several different issues have been faced.

A first important question is, what is timestamped? A few data models timestamp an entire relation, with each version being a separate relation. Most data models timestamp tuples (consider, e.g., TSQL2 [6]), though there several prominent models that timestamp attribute values.

A second question is how is the timestamp represented. Many approaches adopt an interval timestamp, yielding a interval-stamped data model (as in TSQL2 [6]), though the timestamp could also be an instant, yielding a point-stamped data model, or a temporal element (a set of instants, or a set of convex periods).

A third question is whether to support *valid-time* (i.e., the time when data *hold*), *transaction-time* (i.e., the time when data are *inserted\deleted* in the database), or both (bitemporal data). Each kind of time has important wrinkles in its semantics that need to be maintained by the data model.

Additionally, temporal data models may incorporate *temporal integrity constraints* involving current, past, and future database states, and *temporal dependencies* [7]. The concept of "*current time*" is also important. A challenging aspect of supporting this notion of "now" has been to contend with instances that contain this variable when defining the semantics of queries and modification [8] and when supporting queries and updates efficiently, e.g., with the aid of indices. Additionally, several temporal database models have been extended in order to model *temporal indeterminacy* [9], or an explicit representation of *temporal constraints* [10]. Supports have been provided in order to express temporal data at different (and possibly user-defined) *granularities* [11]. Few approaches have also extended the data model (and query language) to support (possibly infinite) *periodical data* (consider, e.g., [12]).

In the area of TDB, also several extensions to the *object-oriented* model have been devised (see, e.g., the survey [13]).

Each data model has a set of operations to *manipulate* and *query* data. A comparative analysis of different temporal algebrae can be found in [14], while an analysis of several SQL-based query languages can be found in [15]. Additionally, temporal logic in database query languages provide expressive query facilities [16].

At the *physical level*, temporal query processing involves disparate architectures, from temporal strata outside the conventional DBMS to adding native temporal support within the DBMS. It is possible to identify three general system architectures that have been used to systematically offer temporal query processing functionality to applications [17]: the *layered* approach uses an off-the-shelf database system and extends it by implementing the missing functionality in a layer between DBMS and applications. The *monolithic* approach integrates the necessary application-specific extensions directly into the database system. The *extensible* approach relies on a database system that allows to plug user-defined extensions into the database system.

Specifically, notice that temporal algebras extend the conventional relational algebra. Some specific operators (e.g., *temporal coalescing* [18], *temporal joins* [19]) have received special attention.

Temporal storage structures and *temporal indexing* have also received a great deal of attention (see, e.g., the comparison in [20]).

To end this part of the section, it is worth highlighting that, although TDB is still an open area of research, many researcher have already consolidated a "basic core" of results, by defining the TSQL2 consensus approach [6]. Interestingly, the core semantics underlying the TSQL2 temporal relational database, i.e., BCDM (Bitemporal Conceptual Data Model) [21], has been proven to be also the semantics at the basis of several other TDB approaches. Recently, Terenziani and Snodgrass [22] have resorted Aristotle's distinction between *telic* and *atelic* events, extending the BCDM semantics (which is suitable only to cope with atelic data, i.e., with data describing some status of the world, such as "*John is employed in the toy department*") to cope also with telic data (e.g., data describing events which have an intrinsic goal or culmination, such has "*ACE built a house*"). A major result of the area is that *upward compatibility* holds with respect to classical (non-temporal) databases: using the BCDM semantics, a temporal database is conceived as a set of classical databases, one for each point in time. Therefore, one can adopt a temporal database without loosing previous work (developed in the context of classical databases). However, despite such a consensus, and crucial properties like upward compatibility, only recently commercial products are starting to include temporal supports.

In general, the problem of extending database to cope with temporal data is task and domain independent. Nonetheless, clinical temporal data show several peculiarities which make their treatment extremely challenging. For instance, Das and Musen have identified several types of *mismatches* between the temporal support of standard databases and the richness of clinical data [23]; analogously, James and Goble have pointed out the requirements that medical records impose on a temporal model [24]. A main issue, for instance, is the fact that durative (interval-timestamped) data must be supported, instead of point-based (single-timestamped) data. Such a problem has been faced, for instance, by Khan and Fagan [25] through the introduction of TNET object oriented database, which has been paired with the temporal query language TQuery [26]. In the relational context, the DXtractor data retrieval application [27] has been used to run interval-based queries onto a standard single-timestamped SQL data. GHC-OSQL [28] is an example of temporal query languages extended in order to support multiple granulatities, besides interval data. On the other hand, Chronus [29] and Chronus II [30] have focused their attention also on the management of temporal data. They are mostly an implementation of a subset of TSQL2 [6], with specific focus on *valid time*, and on *temporal indeterminacy* [31]. On the other hand, Terenziani et al. have proposed an extension of TSQL2 data model in order to cope also with "telic" medical data (such as, e.g., episodes of atrial flutter), which are hardly supported by current TDB approaches [32].

Unfortunately, besides interval-based data, multiple granularities and temporal indeterminacy, a major problem in the treatment of clinical data lies in the gap between "raw" and "abstracted" data. To solve such a problem, several approaches have defined hybrid architectures, in which a Temporal Mediator is used to fulfill the gap. This issue will be faced in section 3.

2. Representing Temporal Information Contained in the Guidelines

A comprehensive comparison among several leading guideline modeling languages can be found in a study by Peleg et al. [33]. In this chapter, we will focus our attention only on the parts of the languages coping with the temporal aspects of clinical guidelines.

The InterMed multiple academic-center (Columbia, Harvard, Stanford) collaboration effort led to the development of a guideline-specification language named the *GuideLine Interchange Format (GLIF)*[34]. GLIF used the *Guideline Expression Language (GEL)* [35] that is based on the Arden Syntax [36]. The GEL temporal expression contains two main types of temporal expression: Times_Expression and Every_Expression. Both of them define the frequency at which an iteration should occur and its duration. A *times expression* specifies that something should occur a specified number of times within a specified interval (e.g., "3 times a day"). An *every expression* specifies that something should occur every *fuzzy duration (e.g.,* every 5 weeks). A *fuzzy duration* is a duration that has an associated before and after uncertainty period.

The GEL language supports temporal relations such as Occurs_at (checks that first argument and the second argument are equal), Is_before (determines whether one date occurs before another), Is_after (determines whether one date occurs after another) or Overlaps (checks whether two intervals overlap).

The GUIDE project [37] is part of a more general framework, Careflow, developed at the University of Pavia, Italy, for modeling and applying clinical guidelines in the broader context of general medical care. The Careflow monitor the patient's history, which is inserted by physicians. For any symptom the physician is required to enter the symptom onset date, time from the onset and the symptom persistence. This information enables the system to used only simple temporal expression such as:" if the time from the onset of symptoms is less than 6 h".

PROforma was developed by Fox et al. [38,39] at the Advanced Computation Laboratory (ACL) of Cancer Research UK. PROforma supports four tasks: actions, plans, decisions, and enquiries. The Plans define sets of tasks which may define logical and temporal constraints (such as follow-up after therapy at ten weeks). In PROforma, guidelines are modelled as plans, and each plan may define constraints on the accomplishment of tasks, as well as task duration and delays between tasks. Moreover, temporal constructs can also be used in order to specify the preconditions of actions. In the PROforma basic modeling reasoning from raw clinical data to yield symbolic, temporal and other abstractions (such as normal/abnormal; rising/falling etc.) is supported.

DILEMMA and PRESTIGE [40] model temporal constraints within conditions. EON [41] uses temporal expressions to allow the scheduling of guideline steps, and deals with duration constraints about activities. Moreover, by incorporating the RESUME system, it provides a powerful approach to cope with temporal abstraction. In EON, the Arden Syntax allows the representation of delays between the triggering event and the activation of a Medical Logic Module (MDL), and between MDLs [42].

A rich ontology to deal with temporal information in clinical trial protocols has been proposed in [43], considering also relative and indeterminate temporal information and cyclical event patterns.

The *Guideline Acquisition, Representation and Execution* (*GLARE*) system [44,45] is a domain-independent system for acquiring, representing and executing clinical guidelines. GLARE's temporal constraint language focuses on the specification of explicit and implicit constraints on the (order of execution of) the actions in the guideline [46,47]. Concerning implicit constraints, the *part-of* relation between component actions and composed ones imposes that the components happen during the action they compose. Concerning explicit constraints, GLARE allows one to express *quantitative* temporal constraints about (possibly indeterminate) *durations* (e.g., *action A must last between 10 and 20 minutes*) and delays (e.g., action *A2 must start no more than 60 minutes after the end of A1*) between actions, and (possibly indeterminate) *qualitative* constraints between endpoints of actions (e.g*., action A2 must be executed after A1*). Specific attention is devoted to the modeling of *repeated* actions. The pattern of repetition can be specified in a recursive way, where arbitrarily many nestings of a specification quadruple of the form *[Nrep, ITime, RConstr, Cond]* can be used. Roughly speaking, the *Nrep* component of the quadruple specifies the number of repetitions, *Itime* represents the time span in which the repetitions must be included, *RConstr* (which is optional) may impose a pattern that the repetitions must follow, and *Cond* (which is optional) allows to express conditions that must hold so that the repetition can take place. Informally, the semantics of a quadruple [Nrep, ITime, RConstr, Cond] can be roughly described by the natural language sentence *"repeat the action Nrep times in exactly ITime, if Cond holds"* (the extensional formal semantics of such a construct is described in [47]). In particular, conditions can have two forms: *while(C)* (where C is a Boolean predicate) stating that, when C becomes false, the repetition ends, and *onlyIf(C)* (where C is a Boolean predicate) stating that, if C is true, the repetition may be performed and, if C is false, the repetition must not be performed and we can pass to the next repetition. For example (assuming the granularity of days) "*Action A is repeated once every week for two weeks, until condition C does not hold anymore*" can be represented using a two-level specification "*repetition(A, [2, 14days, - , while(C)], [1, 7days,- ,-]",* where the inner quadruple (i.e., *[1, 7days,- ,-]*) represents "once every week", while the outermost quadruple represents the "for two weeks" part and the repetition condition.

GLARE allows also to model the time of execution of actions on specific patients, and an explicit *instance-of* relation is used to indicate that a specific action execution is an instance of a generic acion in a guideline. In such a way, instances of actions may "inherit" temporal constraints from the guideline they refer to.

In the Asgaard project [48], a comprehensive conceptual framework for clinical-guideline-based care was developed, as well as an expressive guideline-specification formal language, called Asbru [49], whose focus is on representation of explicit declarative temporal aspects for the guideline process.

The time annotation used within the Asbru language allows a representation of uncertainty in starting time, ending time, and duration of a time interval. The time annotation supports multiple time lines by providing different reference annotations. The *reference annotation* can be an absolute reference point, a reference point with uncertainty (defined by an uncertainty region), a function (e.g., completion time) of a previously executed plan instance, or a domain-dependent time point variable (e.g., CONCEPTION). Temporal shifts from the reference annotation represent uncertainty in the starting time, the ending time, and the overall duration. Thus, the *temporal annotation* represents for each interval the *earliest starting shift* (ESS), the *latest starting shift* (LSS), the *earliest finishing shift* (EFS), the *latest finishing shift* (LFS), the *minimal duration* (MinDu) and the *maximal duration* (MaxDu).

To allow temporal repetitions, the Asbru designers have defined the notion of c*yclical time points* (e.g., MIDNIGHTS, which represents the set of midnights, where each midnight occurs exactly at 0:00 a.m., every 24 hours) and *cyclical time annotations* (e.g., MORNINGS, which represents a set of mornings, where each morning starts at the earliest at 8:00 a.m., ends at the latest at 11:00 a.m., and lasts at least 30 minutes). In addition, certain short-cuts are allowed, such as for the *current time*, whatever that time is (using the symbol *NOW*), or the duration of the plan (using the symbol *). Thus, the Asbru notation enables the expression of interval-based intentions, states, and prescribed actions with uncertainty regarding starting, finishing, duration, and the use of absolute, relative, and even cyclical (with a predetermined granularity) reference annotations.

The actual implementation of the Asbru language requirements for a temporal pattern can be achieved using several languages, such as through the *Constraint based Pattern Specification Language CAPSUL* [50].

3. Temporal Reasoning about Patients' Data: Temporal Abstraction

"The temporal abstraction task can be viewed informally as a type of a generic interpretation task: given a set of time stamped data, external events, and abstraction goals, produce abstractions of the given data that interpret past and present sates and trends and that are relevant for a given set of goals" [51].

In general, the goal of *temporal abstraction* (TA) is to abstract from raw time-stamped input data temporal abstraction useful for a given task (e.g., the execution of a guideline on a given patient). Recently, TA has been generalized to deal also with high-frequency patient streams of data (e.g., deriving from monitoring systems), which, to meet real-time requirements, are not usually stored into Databases (and can be read only once; see, e.g., the survey [52]).

One of the main features of TA, which distinguishes it from Knowledge Discovery and Data Mining in databases, is that TA is mostly a *knowledge-based task*, thus adopting knowledge-based AI techniques rather than pure traditional statistical methods (see, e.g., the discussion in [53]). The input of TA are usually both validated data (data that have usually undergone a process of noise reduction and feature extraction) and a knowledge base (containing both domain knowledge and data abstraction rules), and the output are context-sensitive, qualitative and abstract interval description of such data [52]. In order to better understand the nature of the knowledge base, and the types of TA inferences, we consider, as a seminal example, Shahar's approach [54,51].

In [54,51] four types of knowledge are considered (and represented in a temporal ontology): structural knowledge (e.g., IS-A and PART-OF relations in the domain), (ii) classification knowledge (e.g., classification of Hemoglobin value ranges into LOW, HIGH, VERY HIGH) (iii) temporal semantic knowledge (e.g., the *downward inheritance* property, stating that the value of a given parameter on an temporal interval holds on all subparts of the interval) (iv) temporal dynamic knowledge (e.g., persistence of the value of LOW hemoglobin).

Moreover, in [51] six different subtasks of TA have been identified, which are interesting also in order to illustrate some of the different forms of inferences underlying TA: (i) the context-forming abstraction task, creating relevant frame of reference for patient data interpretation (e.g., the context of AZT drug administration), (ii) vertical temporal inference, creating abstraction by inference from contemporaneous parameters values (e.g., abstracting a Bone-marrow toxicity state from platelet and granulocyte counts), (iii) temporal horizontal inference, inferring a parameter value (e.g., *non-decreasing*) from two other temporally disjoint time values (e.g., *same* and *increasing*), (iv) temporal semantic inference, evaluating the value of parameters over intervals\subintervals on the basis of semantic knowledge (e.g., as an application of the *downward inheritance* property), (v) temporal interpolation, joining sets of disjoint parameters points\intervals into larger intervals including them (e.g., inferring an interval of Low Hemoglobin on the basis of a set of observations of *Low Hemoglobin*), and (vi) temporal pattern matching, inferring complex data patters over time (e.g., inferring an episode of drug toxicity from a state of low white blood cell count lasting more than 2 weeks and starting within 0 to 4 weeks of a state of low Hemoglobin lasting more than 2 weeks in a patient who is receiving certain drugs).

Several excellent surveys have been recently presented concerning TA (consider, e.g., [55,56,52]). In the following, we will briefly mention some of the most interesting approaches.

The seminal approach in [54,51], usually called *knowledge-based temporal-abstraction* (KBTA) *method*, has originally been implemented by the RÉSUMÉ system [54] and since then within multiple other systems, such as the ALMA [57] module with IDAN [58], and evaluated in several clinical domains, such as guideline-based care of oncology and AIDS patients, monitoring of children's growth, and management of patients who have insulin-dependent diabetes. Momentum is a recent extension of the KBTA method for continuous abstraction and monitoring [59]. Thus, a guideline application engine can subscribe to Momentum and be informed of certain temporal abstractions appearing in an individual patient's data or even within a population of patients.

Miksh et al., [60] have provided a comprehensive approach to temporal abstraction, taking into account both point and interval timestamped data to cope with higher frequency observations within data streams (e.g., in the domain of artificial ventilation for neonates). This approach has been implemented in the VIE-VENT system. In [61], considering the general domain of clinical guidelines, such an approach has been further extended to replace the abstraction method in the Asgaard framework [48] to operate in real time with high-frequency data. Recently, Belal et al., [62] have further extended such a work by adopting a fuzzified qualitative trend function to make the system less sensitive when small deviations are tolerable.

Bellazzi et al., [63] employ similar methods of TA as RÉSUMÉ but they also use post-processing of abstracted data, using machine learning and statistical methods to classify data. Interpolation have been studied by several approaches. For instance, Salatian and Hunter [64] have applied a technique using the median filter and rules for temporal inference that create context-sensitive intervals. Larizza et al., [65] have extended classical TA in order to operate in a distributed processing context. Specifically, they have provided an HTTP-based server to carry out TA on behalf of clients used an internet-based infrastructure based on TCP.

An important issue concerning the use of TA in the context of clinical guidelines management is the integration of TA techniques with the Database used in order to store "raw" patient data. By combining the functions of temporal reasoning and temporal maintenance within one architecture, which we refer to as a *temporal mediator* (or, more precisely, a *temporal-abstraction mediator*, since it should include also the data-abstraction capability), a transparent interface can be created, for example from a guideline-application engine, to the patient's time-oriented database.

An example of such an architecture is the Tzolkin temporal-mediation module [66], which supported the EON guideline-based–therapy system [41]. The Tzolkin module combines the RÉSUMÉ temporal-abstraction system [54], the Chronus temporal-maintenance system [29], and a controller into a unified temporal-mediation server. The Tzolkin server supports complex temporal queries, regarding either raw clinical data or their abstractions, submitted by care providers or clinical decision-support applications, hiding the internal division of computational tasks from the user (or from the clinical decision-support application). When users ask complex temporal queries including abstract terms that do not exist in the database, the Tzolkin controller loads the necessary raw data from the database, uses RÉSUMÉ to abstract the data, saves the results in a temporary database, and uses Chronus to access the results and answer the original temporal query.

The IDAN temporal-abstraction architecture [58] has extended the temporal-mediator idea. IDAN has a uniform architecture in which a subset of the temporal- and value-constraints language, the temporal-abstraction rule (TAR) language [57] (which is used in its internal temporal-abstraction computational component, the ALMA system [57]) is used in the query interface of the temporal-abstraction mediator's controller (the process that parses the original query and decides, with the help of the ALMA system, what data and knowledge should be used). Thus, ALMA can also process the query's temporal constraints. Unifying both tasks avoids re-implementation of the constraint-satisfaction process and the use of a temporary storage space. ALMA in act implements through its TAR rules the knowledge-based temporal-abstraction method [54,51].

IDAN is also fully distributed and can access multiple clinical databases, medical knowledge bases, and, in theory, multiple computational temporal-abstraction modules. An IDAN session starts by defining a particular data, knowledge, and processing configuration and then referring raw-data or abstract-concept queries to the controller [58]. IDAN is used by multiple applications, from patient data exploration in the KNAVE-II system [67] to answering the guideline application engine's queries during guideline-based care. An example is *DeGeL* [68], a distributed framework that supports clinical-guideline specification, retrieval, application, and quality assessment, by sending runtime queries about the current patient to the IDAN controller through its runtime application module.

Recently, the mediator methodology has also been used to couple Chronus II [30] with a temporal ontology (specified using OWL) and a temporal pattern specification language supporting data abstraction [69].

4. Temporal Reasoning about Guidelines: Constraint Propagation

Temporal reasoning about constraints is a widely explored area of research in Artificial Intelligence: in this section, we first briefly overview some of the main concepts and results, and then focus on their application to the clinical guideline context.

Roughly speaking, one could distinguish between two different mainstreams in the research about time carried on within the AI community: "general-purpose" and "constraint-based" approaches. General-purpose approaches (which are not coped with in this section) are *logical* approaches, mainly focusing on the definition of a formalism general enough to represent the dynamic aspects of the world, and adopting theorem proving to perform temporal reasoning (consider, e.g., [70,71]). On the other hand constraint-based approaches mainly focus on the definition of a representation formalism and of reasoning techniques to deal specifically with temporal constraints between temporal entities (time points and/or time intervals) per se, independently of the events and states which take place over such entities. For instance, given three time intervals I1, I2 and I3, if I1 is before I2 and I2 is before I3, then one can infer that I1 is before I3, independently of the events that occurred in I1, I2 and I3. By focusing on a more restricted problem, and with a careful definition of the temporal constraint language, one can define specialised *constraint propagation techniques* that make inferences such as the above one in a more efficient way than, e.g., a standard theorem prover for the first-order logic. As a consequence, the analysis of the *trade-off* between the *expressiveness* of the constraint language and the *computational complexity* of the *correct* and *complete* constraint propagation techniques operating on them is a central issue within this mainstream of research (consider, e.g., the surveys [72,73,74]).

While expressiveness is an obvious desideratum, we will now briefly motivate the second term of the above trade-off. First, it is important to stress that formalisms for temporal constraints are not very useful if they are not paired with constraint propagation algorithms. Consider, e.g., a Knowledge Base KB containing the temporal constraints (i) and (ii) between three events A, B, and C.

KB= {(i) *A before B*; (ii) *B before C*}

The constraint (iii) *A before C* can be inferred (i.e., it is logically implied by (i) and (ii)), so that, given KB, one can correctly assert (iii), but not (iv) *A after C*, which is actually inconsistent with KB (in other words, the set of constraints KB'={(i), (ii), (iv)} cannot be satisfied). Temporal constraint propagation is necessary in order to support such an intended semantics. With no temporal reasoning, a user can represent any set of constraints, even an inconsistent one (e.g., KB' above) with no reaction by the system. Of course, temporal reasoning algorithms are computationally expensive. An important desideratum is *tractability*, i.e. the fact that the running time of the algorithms grows as a fixed power of the number of the actions and/or constraints in the knowledge base (i.e., polynomial time). However, temporal constraint propagation algorithms should also be *correct*, i.e., such that they only infer constraints that are logically implied by the initial set of constraints (correctness grants that no wrong inference is made). *Completeness* (i.e., the fact that all logically implied constraints are actually inferred) is a fundamental desideratum as well, since it is essential in order to grant that the system's answers are fully reliable (e.g., if (iii) is not inferred from {(i), (ii)}, the fact that the set of constraints KB'={(i), (ii), (iv)} is inconsistent is not detected by the system.

To wrap up the above discussion, and focusing on the guideline context, it is worth stressing that, in presence of temporal constraints, *the only way to be sure that a guideline is temporally consistent (and, therefore, executable!) is to have a correct and complete inferential mechanism (e.g., in the form of a constraint propagation algorithm) operating on the temporal constraints it contains*.

In the area of AI, several temporal constraint languages have been proposed, and constraint propagation algorithms devised. Concerning *qualitative* temporal constraints (i.e., constraints on the relative order of events), the most famous approach is probably Allen's Inteval Algebra [75] (henceforth IA). Allen identified 13 possible base relations, coding the different possible positions of two time intervals. Disjunctions of base relations (e.g., I1 BEFORE or DURING I2) are used to represent uncertain cases. Allen's constraint propagation algorithm operates in a time cubic on the number of time intervals. However, such an algorithm is not complete for IA. While many approaches chose to adopt Allen's algorithm, other approaches tried to design less expressive tractable formalisms. For example, the Point Algebra (henceforth: PA) is defined in the same way as IA, but the temporal elements are time points [76]. Thus, there are only three primitive relations between time points (i.e., <,=, and >), and four ambiguous relations (i.e., (<,=), (>,=), (<,>), and (<,=,>)). In the Point Algebra, the constraint closure can be computed in polynomial time by algorithms that are both sound and complete. Obviously, the price to be paid for tractability is the expressive power: not all relations between time intervals can be mapped onto relations between their endpoints. An interesting algebra is the Continuous Point Algebra (CPA), which consists of all relations in the Point Algebra excluding inequality (i.e., excluding the ambiguous relation (<,>)). In fact, Allen's path consistency algorithm is both sound and complete for such an algebra (for more details, see, e.g., the surveys in [72,73,74]). A different simplification of Allen's Algebra has been provided by Freksa [77]. Freksa has identified coarser qualitative temporal relations than Allen's ones, based on the notion of semi-intervals (i.e., beginnings and ending points of durative events). Freksa has also shown that relations between semi-intervals result in a possible more compact notation and more efficient reasoning mechanisms, in particular if the initial knowledge is, at least in part, coarse knowledge.

Another mainstream of research about *qualitative temporal reasoning* focused on the identification of tractable fragments of Allen's algebra. The milestone work by Nebel and Burkert [78] first pointed out the "ORD-Horn subclass", showing that reasoning in such a class is a polynomial time problem and that it constitutes a maximal tractable subclass of Allen's algebra.

Quantitative temporal constraints involve *metric* time and include dates (e.g., "John arrived on 10/10/99 at 10:00"), durations (e.g., "John worked for 3 hours") and delays (e.g., "John arrived 10 minutes after Mary"). In many cases, metric temporal information is not so precise: one can have approximate dates (see, e.g., Ex.1), durations (Ex.2), and distances (delays) between time points and/or endpoints of time intervals (Ex.3).

(Ex.1) the time interval I1 ended on 10/10/99, between 10:00 and 10:15

(Ex.2) I1 lasted between 20 and 30 minutes

(Ex.3) I2 started between 20 and 40 minutes after the end of I1

Temporal constraint propagation is important in order to infer new temporal constraints and to detect inconsistencies. For instance, from the constraints in Ex.1, Ex.2 and Ex.3 one can infer that I2 started on 10/10/99, between 10:20 and 10:55, 40—70 minutes after the start of I1, so that the set of constraints {Ex.1, Ex.2, Ex.3, Ex.4} is inconsistent.

(Ex.4) I2 started at 10:10

Of course, problems become more complex in case also disjunctions of temporal constraints (see, e.g., Ex.5) are taken into account

(Ex.5) I2 started 20-30 or 50-60 minutes after the end of I1

Many AI approaches have been developed in order to face some or all of the above issues. For instance, Dechter et al. [79] have proposed a model (called Temporal Constraint Satisfaction Problem – TCSP) based on the primitive notions of time points and distances between time points, which allows one to cope with constraints such as Ex.1—5 above. Dechter et al. developed an optimized constraint propagation algorithm to perform temporal reasoning taking advantage of a graph representation of the constraints, that operates in exponential time. On the other hand, in the case disjunctions of distances are not allowed the resulting constraint problem (called Simple Temporal Problem; henceforth: STP) can be solved in a time cubic on the number of nodes (time points) using a standard all-to-all shortest path algorithm (e.g., Floyd Warshall's algorithm), which is both correct and complete for STP. Finally, notice that constraints in Exs 1—4 (but not Ex.5) can be mapped onto a STP.

Furthermore, several *integrated* approaches have been devised in order to deal with both qualitative and quantitative temporal constraints. For example, in the LaTeR temporal manager [80], the high-level language allows one to deal with both time points and time intervals, and to express both quantitative and qualitative temporal constraints. As a further example, Jonsson and Backstrom [81] proposed an homogeneous framework, based on linear programming, that deals with all the types of constraints discussed above, and that also allows one to express constraints on the relative duration of events.

Other Artificial Intelligence approaches have focused on the treatment of temporal constraints concerning *repeated* and possibly *periodic* actions (consider, e.g., [82,83,84]). For instance, Terenziani [84] has proposed an extension of Allen's algebra to consider qualitative relations between periodic facts. Terenziani's approach deals with constraints such as (Ex.6):

(Ex.6) Between January 1, 1999 and December 31, 1999 on the first Monday of each month, Andrea went to the post office before going to work.

In [84], temporal reasoning over such constraints is performed by a path consistency algorithm which extends Allen's one. Such an algorithm is sound but not complete and operates in cubic time with respect to the number of periodic facts.

Recent developments also include incremental [85] and fuzzy [86] temporal reasoning, and optimized treatment of Disjunctive Temporal Problems [87,88].

A relatively less explored area of research, which is, on the other hand, of primary importance in clinical guidelines applications, is the integration of temporal constraint propagation techniques with the temporal (relational) database technology. Koubarakis [89] first extended the *constraint database model* to include indefinite (or uncertain) temporal information (including qualitative temporal constraints). Koubarakis proposed an explicit representation of temporal constraints on data; moreover, the local temporal constraints on tuples are stored into a dedicated attribute. He also defined the algebraic operators, and theoretically analysed their complexity. On the other hand, the work by Brusoni et al., [10] mainly focused on defining an integrated approach in which "standard" Artificial Intelligence temporal reasoning capabilities (such as the ones sketched above in this entry) are suitably extended and paired with an (extended) relational temporal model. First, the data model is extended in such a way that each temporal tuple can be associated with a set of identifiers, each one referring to a time interval. A separate relation is used in order to store the qualitative (and quantitative) temporal constraints about such intervals. The algebraic operations of intersection, union and difference are defined over such sets of periods, and indeterminacy (e.g., about the existence of the intersection between two periods) is coped with through the adoption of conditional intervals. Algebraic relational operators are defined on such a data model, and their complexity analysed. Finally, an integrated and modular architecture combining a temporal reasoner with an extended temporal database is described, as well as a practical application to the management of temporal constraints in clinical protocols and guidelines.

In the area of clinical guidelines, the development of temporal constraint propagation algorithms has not attracted a lot of attention. This is, actually, a quite severe drawback for the area. As a matter of facts, temporal constraints in the guidelines are often very complex: they are usually both qualitative and quantitative, and may involve not only temporal indeterminacy, but also possibly periodic patterns of repetitions. And, given such a complexity, no current AI "standard" constraint propagation approach is expressive enough to cope with all of such constraints in an integrated way. The two most notable exceptions are the constraint propagation approaches devised by Duftschmid et al. [90] and by Terenziani et al., [46,47].

Duftschmid et al. have proposed a comprehensive temporal constraint propagation approach based on the temporal language of Asbru (see section 2). In particular, in Duftschmid et al.'s approach, different types of temporal constraints – deriving from the scheduling constraints in the guideline, from the hierarchical decomposition of actions into their components and from the control-flow of actions in the guideline – are mapped onto an STP framework [79]. Temporal constraint propagation is used in order to (1) detect inconsistencies, and to (2) provide the minimal constraints between actions. In [90], there is also the claim that (3) such a method can be used by the guideline interpreter in order to assemble feasible time intervals for the execution of each guideline activity. Moreover, advanced visualization techniques are used in order to show users the results of temporal reasoning (see chapter 8 of this book).

While in [90] the *completeness* of the constraint propagation algorithms is not discussed, the definition of a *tractable*, *correct* and *complete* constraint propagation algorithm has been the core goal of the approach by Terenziani et al. [46,47], which operates on the temporal constraint language of GLARE (see section 2). As a matter of facts, GLARE's temporal language has been carefully designed considering the expressiveness vs. computational complexity trade-off. For instance, in GLARE's temporal language, different forms of temporal indeterminacy are allowed, but no indeterminacy can be stated about the duration of (possibly periodic) repeated actions, since such a form of indeterminacy would make the correct and complete temporal constraint propagation algorithm exponential. In order to perform temporal reasoning, GLARE's temporal constraints are mapped onto an internal data structure, called *STP-tree* [46], which extends standard STP [79] to cope with periodic and repeated actions. A separate STP is used in order to model patients' temporal data. Two basic constraint propagations algorithms are devised in order to check the consistency of the temporal constraints in a guideline (taken in isolation) and to check whether the time of execution of actions on the patient (taken from the patients' data) is consistent with such constraints. Both algorithms are proven to be correct and complete, and to operate in a time cubic in the number of actions.

To conclude this section, it is worth pointing out and stressing the main advantages that the adoption of a temporal constraint propagation approach can contribute to a computer-based manager of temporal guidelines. Specifically, at least four types of facilities can be provided (see e.g. [47] for a more detailed analysis):

1. the consistency-checking-guideline facility: this facility can be used in order to check the temporal consistency of the guideline in a principled way. Such a facility can be advocated at any stage during the acquisition of a clinical guideline, so that incremental consistency checking is also possible.
2. the consistency-checking-instance facility: this facility can be used in order to check whether the temporal constraints in the guideline have been respected or not by the instances of actions that have been executed on the specific patients (considering also partial - i.e., ongoing executions).

3. the query facilities: during the execution of a given guideline, the query facilities provide the user-physicians with a tool to obtain temporal information. This temporal information have often a crucial role when the user-physicians must take a decision.
4. the next-action facility: given the temporal constraints in the whole guideline and given the time when the last actions in the guideline have been executed, determining the set of next candidate actions and the window of time when they need to be executed is a complex task, which requires temporal constraint propagation

5. Research Agenda

The International Workshop on Computer-Based Clinical Guidelines and Protocols in Leiden, Netherlands, 9-11 January 2008, has been a unique opportunity for the clinical guideline community to meet and discuss in detail the status of the art, and to try to delineate an agenda of future research in the area. Concerning the treatment of temporal aspects, the following main tasks have been identified in the Workshop:

1. Treatment of multiple granularities, and of temporal indeterminacy, at the level of both patients' data and guidelines. As a matter of fact, although several approaches in the artificial intelligence, temporal database and clinical guidelines research areas have already been devised to cope with multiple granularities and\or temporal indeterminacy, this issue is still an open problem, which needs substantial further research, as well as a deeper integration of the results independently obtained in the aforementioned areas of research;
2. Combination of patient temporal data with guidelines. This is a core issue as regards both temporal abstraction and temporal constraint propagation, which strictly relates also the aforementioned treatment of temporal indeterminacy and multiple granularities. For instance, combining temporal indeterminacy at the patients' data level with temporal indeterminacy in the guideline is still a challenging open issue;
3. Development of "layered" temporal approaches. The research in artificial intelligence has widely demonstrated that the complexity of temporal inference is strictly related to the expressiveness of temporal languages. Therefore, a "layered" approach, providing different levels of temporal languages and inferential capabilities, to give users the possibility of choosing the level best suited for their specific application, would provide a fundamental enhancement of the state of the art, with a major impact on the applicability of the techniques being devised;
4. Development of proper interface tools, to acquire temporal data and\or communicate it to guideline users. Although several valuable visualization techniques have been already devised (see chapter 8 of this book), the treatment of temporally indeterminate data, and of multiple granularities, arise challenging problems as regards man-machine interaction;
5. Identification of next candidate actions. The identification of the next actions to be executed is one of the primary tasks of the execution engine of guideline systems. However, in the presence of temporal constraints and temporal

indeterminacy, such an identification requires complex interactions between temporal constraint propagation algorithms and the execution engine, which have only been partially explored until now;

6. Treatment of low/high frequency domains. As mentioned in section 3, patient data may be monitored at very different frequencies, so that different techniques may be needed to store and analyse them (e.g., multiple data stream analysis may be required for high-frequency monitoring data).

References

[1] J. Chomicki: Temporal query languages: a survey, in *Proc. of Temporal Logic: ICTL'94*, vol. **827**, D. M. Gabbay and H. J. Ohlbach, eds, Springer-Verlag (1994), 506–534.

[2] Gultekin, Ozsoyoglu and R.T. Snodgrass: Temporal and Real-Time Databases: A Survey, *IEEE Transactions on Knowledge and Data Engineering* **7(4)** (1995), 513–532.

[3] Y. Wu, S. Jagodia, and X. S. Wang: Temporal Database Bibliography Update, Temporal Databases—Research and Practice*,* Springer-Verlag, *LNCS* **1399** (1999), 338–367.

[4] P. Chen: The Entity-Relationship Model - Toward a Unified View of Data, *ACM Transactions of Database Systems* **1(1)** (1976), 9-36.

[5] H. Gregersen and C. S. Jensen: Temporal Entity-Relationship Models-A Survey, *IEEE Transactions on Knowledge and Data Engineering* **11(3)** (1999), 464-497.

[6] R. T. Snodgrass (Editor): *The TSQL2 Temporal Query Language*, Kluwer Academic Publishers (1995), 674+xxiv pages.

[7] C. S. Jensen, R. T. Snodgrass, and M. D. Soo: Extending existing dependency theory to temporal databases, *IEEE Transactions on Knowledge and Data Engineering* **8(4)** (1996), 563–582.

[8] K. Torp, C. S. Jensen and R. T. Snodgrass: Modification Semantics in Now-Relative Databases, *Information Systems* **29(78)** (2004), 653–683.

[9] C. Dyreson and R. T. Snodgrass: Supporting Valid-time Indeterminacy, *ACM Transactions on Database Systems* **23(1)** (1988), 1-57.

[10] V. Brusoni, L. Console, B. Pernici, and P. Terenziani: Qualitative and Quantitative Temporal Constraints and Relational Databases: Theory, Architecture, and Applications, IEEE *Transactions on Knowledge and Data Engineering* **11(6)** (1999), 948-968.

[11] C. Bettini, X. S. Wang, and S. Jajodia: *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer Verlag, (2000).

[12] P. Terenziani: Symbolic User-defined Periodicity in Temporal Relational Databases, *IEEE Transactions on Knowledge and Data Engineering* **15(2)** (2003) 489-509.

[13] R.T. Snodgrass: Temporal object-oriented databases: A critical comparison, *Modern Database Systems*, (1995) 386–408.

[14] E. McKenzie, and R. T. Snodgrass: An Evaluation of Relational Algebras Incorporating the Time Dimension in Databases, *ACM Computing Surveys* **23(4)** (1991), 501–543.

[15] A.U. Tansel, and E. Tin: Expressive Power of Temporal Relational Query Languages, *IEEE Transactions on Knowledge and Data Engineering* **9(1)** (1997), 120-134.

[16] J. Chomicki, D. Toman, and M. H. Bohlen: Querying ATSQL Databases with Temporal Logic. *ACM Transactions on Database Systems* **26(2)** (2001), 145–178.

[17] M. Koubarakis, T. K. Sellis, A. U. Frank, S. Grumbach, R. H. G¨uting, C. S. Jensen, N. A. Lorentzos, Y. Manolopoulos, E. Nardelli, B. Pernici, H. Schek, M. Scholl, B. Theodoulidis, and N. Tryfona (Editors): Spatio-Temporal Databases: The CHOROCHRONOS Approach, **LNCS 2520**, Springer, (2003).

[18] M. H. Bohlen, R. T. Snodgrass, and M. D. Soo: Coalescing in Temporal Databases. Proceedings of the International Conference on Very Large Data Bases, Morgan Kaufmann Publishers, Mumbai (Bombay), India, (1996), 180–191.

[19] D. Gao, R. T. Snodgrass, C. S. Jensen and M. D. Soo: Join operations in temporal databases. *VLDB Journal* **14(1)** (2005), 2–29.

[20] B. Salzberg and V.J. Tsotras: A Comparison of Access Methods for Time-Evolving Data. *ACM Computing Surveys* **31(2)** (1999), 158-221.

[21] C. S. Jensen and R. T. Snodgrass: Semantics of Time-Varying Information, *Information Systems* **21(4)** (1996), 311–352.

[22] P. Terenziani, R.T. Snodgrass: Reconciling Point-based and Interval-based Semantics in Temporal Relational Databases: A Proper Treatment of the Telic/Atelic Distinction, *IEEE Transactions on Knowledge and Data Engineering* **16(4)** (2004), 540-551.

[23] A.K. Das, M.A. Musen: A foundational model of time for heterogeneous clinical databases, Proc. *AMIA'97* (1997), 106-110.

[24] R. James, C. Goble: Survey and critique of time and medical records, Proc. *Medinfo'95* (1995), 271-275.

[25] M.G. Kahn, L.M. Fagan, and S. Tu: Extensions to the time-oriented database model to support temporal reasoning in medical expert systems, *Methods Inf. Med* **30(1)** (1991) 4-14.

[26] M.G. Kahn, S. Tu, and L.M. Fagan: TQuery: a context-sensitive temporal query language, *Comput. Biomed. Res.* **24(5)** (1991), 401-419.

[27] D.J. Nigrin, and I.S. Kohane: Temporal Expressiveness in Querying a Time-stamp-based Clinical Database, *Journal of the American Medical Informatics Association* **7** (2000), 152-161.

[28] C. Combi, G. Cucchi, and F. Pinciroli: Applying object-oriented technologies in modeling and querying temporally oriented clinical databases dealing with temporal granularity and indeterminacy, *IEEE Trans. Information Technology in Biomedicine* **1(2)** (1997), 100-127.

[29] A.K. Das, and M.A. Musen: A temporal query system for protocol-directed decision support, *Methods Inf. Med.* **33(4)** (1994) 358-70.

[30] M.J. O' Connor, S. Tu, M.A. Musen: The Chronus II Temporal Database Mediator, Proc. *AMIA'02*, (2002), 567-571.

[31] M.J. O' Connor, S. Tu, and M.A. Musen: Representation of Temporal Indeterminacy in Clinical Databases, Proc. *AMIA'00* (2000), 615-619.

[32] P. Terenziani, R.T. Snodgrass, A. Bottrighi, M. Torchio, and G. Molino: Extending temporal database to deal with telic/atelic medical data, *Artificial Intelligence In Medicine* **39(2)** (2007) 113-126.

[33] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E.H. Shortliffe, and M. Stefanelli: Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *JAMIA* **10(1)** (2002) 52–68.

[34] Peleg, M., Boxwala, A. A., Omolola, O., Zeng, Q., Tu, S.W., Lacson, R., Bernstam, E., Ash, N., Mork, P., Ohno-Machado, L., Shortliffe, E.H., and Greenes, R.A.: GLIF3: The Evolution of a Guideline Representation Format, Proc. *AMIA Annual Symposium* (2000).

[35] Ogunyemi, O., The Guideline Expression Language (GEL) User's Guide. Technical Report. Boston, MA: Brigham and Women's Hospital; Report No.: DSG-TR-2000-001 (2000).

[36] Peleg, M., Ogunyemi, O., Tu, S.W., Boxwala, A.A., Zeng, Q., Greenes, R.A., et al. Using features of Arden syntax with object-oriented medical data models for guideline modeling, Proc A*MIA'01* (2001).

[37] Quaglini, S., Stefanelli, M., Lanzola, G., Caporusso, V., Panzarasa, S.: Flexible guideline-based patient careflow systems. *Artificial Intelligence in Medicine* **22(1)** (2001) 65-80.

[38] Fox, J., Johns, N., and Rahmanzadeh, A: Disseminating medical Knowledge: the PROforma approach, *Artificial Intelligence in Medicine* **14** (1998), 157-181.

[39] Sutton, D.R., Fox, J.: The syntax and semantics of the PROforma guideline modeling language. *Journal of the American Medical Informatics Association* **10(5)** (2003), 433–443.

[40] C. Gordon, S.I. Herbert, P. Johnson: Knowledge Representation and Clinical Practice Guidelines: the DILEMMA and PRESTIGE projects, *Medical Informatics Europe '96*, IOS Press, (1996), 511-515.

[41] M.A. Musen, S.W. Tu, A.K. Das, and Y. Shahar: EON: A component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Information Association* **3(6)** (1996), 367-388.

[42] E. Sherman, G. Hripcsak, J. Starren, R. Jender, and P. Clayton: Using intermediate states to improve the ability of Arden Syntax to implement care plans and reuse knowledge, Proc. *Annual Symposium on Computer Applications in Medical Care* (1995), 238-242.

[43] C. Weng, M. Kahn, J.H. Gennari: Temporal Knowledge Representation for Scheduling Tasks in Clinical Trial Protocols, Proc. *AMIA'02*, San Antonio TX (2002), 879 - 883.

[44] P. Terenziani, G. Molino, and M. Torchio: A Modular Approach for Representing and Executing Clinical Guidelines, *Artificial Intelligence in Medicine* **23** (2001) 249-276.

[45] P. Terenziani, S. Montani, A. Bottrighi, G. Molino, and M. Torchio: Applying Artificial Intelligence to Clinical Guidelines: the GLARE Approach, in *Computer-based medical guidelines and protocols: A primer and current trends*, A ten Teije, S Miksch and P Lucas (Eds.), IOS Press, 2008.

[46] P. Terenziani, S. Montani, M. Torchio, G. Molino, and L. Anselma: Temporal Consistency Checking in Clinical Guidelines Acquisition and Execution: the GLARE's Approach, *Proc. Fall AMIA Symposium*, (2003), 659-663.

[47] L. Anselma, P. Terenziani, S. Montani, and A. Bottrighi: Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines, *Artificial Intelligence in Medicine* **38(2)** (2006) 171-195.

[48] Shahar, Y., Miksch, S., and Johnson, P.: The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine* **14**, (1998), 29-51.

[49] Miksch, S., Shahar, Y., Johnson, P.: Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. Proc. of the seventh workshop on knowledge engineering methods and languages (*KEML-97*), Milton Keynes, UK, (1997).

[50] Chakravarty, S., Shahar, Y.: CAPSUL- A constraint-based specification of repeating patterns in time-oriented data. *Annals of Mathematics and Artificial Intelligence* **30** (2000), 3-22.

[51] Y. Shahar: A framework for knowledge-based temporal abstraction", *Artificial Intelligence* **90** (1997), 79-133.

[52] M. Stacey and C. McGregor: Temporal abstraction in intelligent clinical data analysis: A survey, *Artificial Intelligence in Medicine* **39** (2007), 1-24.

[53] N. Lavrac, E. Keravnou, B. Zupan: Intelligent data analysis in medicine. In A. Kent (ed.) *Encyclopaedia of Computer Science and Technology* **42**. New York, Marcel Dekker (2000), 113-157.

[54] Y. Shahar, M.A. Musen: Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine* **8(3)** (1996), 267-298.

[55] C. Combi, and Y. Shahar: Temporal reasoning and temporal data maintenance in medicine: issues and challenges, *Computational Biology in Medicine* **27** (1997), 353-368.

[56] J.C. Augusto: Temporal reasoning for decision support in medicine, *Artificial Intelligence in Medicine* **33** (2005), 1-24.

[57] Balaban, M., Boaz, D., and Shahar, Y.: Applying temporal abstraction in medical information systems. *Annals of Mathematics, Computing, & Teleinformatics* **1(1)** (2003), 54-62.

[58] D. Boaz, and Y. Shahar: Idan: A Distributed Temporal-Abstraction Mediator for Medical Databases. Proc. *9th Conf. on Artificial Intelligence in Medicine—Europe (AIME'03)*, Protaras, Cyprus (2003).

[59] A. Spokoiny, Y. Shahar: A Knowledge-Based Time-Oriented Active Database approach for Intelligent abstraction, Querying and Continuous Monitoring of Clinical Data, Proc. *Medinfo'04* (2004), 84-88.

[60] S. Miksch, W. Horn, and C. Popow: Context-sensitive and expectation guided temporal abstraction of high-frequency data, Proc. *10th International Workshop for Qualitative Reasoning (QR'96)*, Menlo Park, AAAI press (1996), 154-163.

[61] A. Seyfang, S. Miksch: Advanced temporal data abstraction for guideline execution, Proc. *Symposium on Computerized Guidelines and Protocols (CGP 2004)*. IOS Press (2004), 88-103.

[62] S.Y. Belal, A.F.G. Taktak, A. Nevill, A. Spencer: An intelligent ventilation and oxygenation management system in neonatal intensive care using fuzzy trend template fitting, *Pysiological Measures* **26** (2005), 555-570.

[63] R. Bellazzi, C. Larizza, P. Magni, S. Montani, and M. Stefanelli: Intelligent analysis of clinical time series: an application in the diabetes mellitus domain, *Artificial Intelligence in Medicine* **20** (2000), 37-57.

[64] A. Salatian, and J. Hunter: Deriving trends in continuously sampled medical data, *Journal of Intelligent Information Systems* **13** (1999), 47-71.

[65] C. Larizza, R. Bellazzi, and G. Lanzola: An http-based server for temporal abstractions, Proc. *IDAMAP'99*, (1999).

[66] Nguyen, J., Shahar, Y., Tu. S.W., Das, A.K., and Musen, M.A.: Integration of temporal reasoning and temporal-data maintenance into a reusable database mediator to answer abstract, time-oriented queries: The Tzolkin System. *Journal of Intelligent Information Systems* **13 (1/2)** (1999), 121-145.

[67] Martins S.B., Shahar Y., Galperin M., Kaizer H., Goren-Bar D., McNauughton D., Basso L.V., Goldstein M.K.: Evaluation of KNAVE II: a tool for intelligent query and exploration of patient data, Proc. *Medinfo'04* (2004), 648-652.

[68] Shahar Y, Young O., Shalom E., Mayaffit A., Moskovitch R., Hessing A., and Galperin M.: DEGEL: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines. Proc. 9th *Conference on Artificial Intelligence in Medicine—Europe (AIME'03*), Protaras, Cyprus, (2003).

[69] M.J. O'Connor, R.D. Shankar, D.B. Parrish, and A.K. Das: Knowledge-level Querying of Temporal Patterns in Clinical Reasearch Systems, Proc. *Medinfo'07* (2007), 311-315.

[70] J. Van Benthem: The logic of time, Kluwer Academic, Dordrect, (1983).

[71] A.E. Emerson: Temporal and Modal Logic, in: Van Leeuwen (Editor) *Handbook of Theoretical Computer Science* **B**, Elsevier Science Publishers, (1990), 997-1072.

[72] Vilain M, Kautz H, and VanBeek P.: Constraint Propagation Algorithms for temporal reasoning: a revised report. in: D.S. Weld and J. deKleer (edsEditors) *Readings in Qualitative Reasoning about Physical Systems*, Morgan Kaufmann (1990), 373-381.

[73] J. Allen: Time and time again: the many ways to represent time, *Intelligent Systems* **6(4)** (1991), 341-355.

[74] L. Vila: A survey on temporal reasoning in artificial intelligence, *AI Communications* **7(1)** (1994), 4-28.

[75] J. Allen: Maintaining knowledge about temporal intervals, *Communications of the ACM* **26(11)** (1983), 832-843.

[76] Vilain, M., and Kautz, H.: Constraint propagation algorithms for temporal reasoning, Proc. *Fifth National Conference on Artificial Intelligence (AAAI'86)*, Philadelphia, PA (1986), 377-382.

[77] C. Freksa: Temporal reasoning based on semi-intervals. *Artificial Intelligence* **54(1-2)** (1992), 199-227.

[78] Nebel, B., and Burkert, H.J.: Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra, *Journal of the ACM* **42(1)** (1995), 43-66.

[79] Dechter R, Meiri I, and Pearl J.: Temporal Constraint Networks, *Artificial Intelligence* **49** (1991), 61-95.

[80] Console L. and Terenziani P.: Efficient processing of queries and assertions about qualitative and quantitative temporal constraints, *Computational Intelligence* **15(4)** (1999), 442-465.

[81] Jonsson P. and Backstrom C.: A unifying approach to temporal constraint reasoning, *Artificial Intelligence* **102** (1998), 143-155.

[82] P. Ladkin: Time Representation: A Taxonomy of Interval Relations, Proc. *fifth National Conf. on Artificial Intelligence*, Philadelphia, PA (1986), 360-366.

[83] R.A. Morris, W.D. Shoaff, and L. Khatib: Domain Independent Temporal Reasoning with Recurring Events, *Computational Intelligence* **12(3)** (1996), 450-477.

[84] Terenziani P.: Integrating calendar-dates and qualitative temporal constraints in the treatment of periodic events, *IEEE Transactions on Knowledge and Data Engineering* **9(5)** (1997), 763-783.

[85] Gerevini, A.: Incremental qualitative temporal reasoning: algorithms for the point algebra and the ORD-Horn class, *Artificial Intelligence* **166(1-2)** (2005), 37-80.

[86] Badaloni, S., Giacomin, M.: The algebra IAfuz: a framework for qualitative fuzzy temporal reasoning, *Artificial Intelligence* **170(10)** (2006), 872-908.

[87] K. Stergiou and M. Koubarakis: Backtracking algorithms for disjunctions of temporal constraints. Proc. *15th National Conference on Artificial Intelligence (AAAI-98)* (1998), 248–253.

[88] I. Tsamardinos and M. E. Pollack. Efficient solution techniques for disjunctive temporal reasoning problems. *Artificial Intelligence* **151(1-2)** (2003), 43–89.

[89] M. Koubarakis: Database models for infinite and indefinite temporal information, *Information Systems* **19(2)** (1994), 141-173.

[90] G. Duftschmid, S. Miksch, W. Gall: Verification of temporal scheduling constraints in clinical practice guidelines. *Artificial Intelligence in Medicine* **25(2)** (2002), 93-121.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-101

Chapter 6
Planning: Supporting and Optimizing Clinical Guidelines Execution

Luca ANSELMA[a] and Stefania MONTANI[b,1]
[a] *Dipartimento di Informatica, Università di Torino, Torino, Italy*
[b] *Dipartimento di Informatica, Università del Piemonte Orientale, Alessandria, Italy*

Abstract. A crucial feature of computerized clinical guidelines (CGs) lies in the fact that they may be used not only as conventional documents (as if they were just free text) describing general procedures that users have to follow. In fact, thanks to a description of their actions and control flow in some semiformal representation language, CGs can also take advantage of Computer Science methods and Information Technology infrastructures and techniques, to become *executable documents*, in the sense that they may support clinical decision making and clinical procedures execution. In order to reach this goal, some advanced planning techniques, originally developed within the Artificial Intelligence (AI) community, may be (at least partially) resorted too, after a proper adaptation to the specific CG needs has been carried out.

Keywords. Planning, Artificial Intelligence, time, resources, uncertainty, indeterminacy, optimized CG execution, adaptation, decision support

Introduction

The area of planning in AI has been considerably developed during the last few years, and it is closely related to the topic of execution of CGs. In the first part of this chapter, a description of the state of the art of planning in AI, of the issues that have been already addressed and of how they were managed is provided. In fact, although AI researchers will probably need to engage in further investigation in order to provide planning techniques directly applicable to the execution of CGs, it may be useful to have an overview of the approaches originally devised by the AI planning community. After illustrating the peculiarity of the CG domain and its impact on the definition of a planning activity, the second part of the chapter presents different planning tasks that are being dealt with by scientists belonging to the community dedicated to CG computerized management research, and discusses how advanced AI planning techniques can be adapted and then (partially) relied upon to this end. This section is organized according to the specific activities that have to be performed at CG execution time; specifically, we deal with plan modification, plan execution in the strict sense, plan critiquing, and plan visualization.

[1] Corresponding Author: DI, Università del Piemonte Orientale "Amedeo Avogadro", Via Bellini 25/g, 15100 Alessandria, Italy; E-mail: stefania.montani@unipmn.it.

1. Planning in Artificial Intelligence

The goal of this section is to introduce planning approaches in AI, as well as problems they engender and solutions, either already achieved or currently explored. This section is not meant to be exhaustive, but rather to address central issues related to planning in AI. Therefore, an accessible presentation of the topic is provided, focusing on the general ideas in the area of research on planning, rather than on specific features provided by specific planners. The interested reader may refer, for example to [1] for an in-depth treatment. Please note that, in accordance to the nature of this book, we introduce planning at a high level suitable to readers not expert in AI, and we deliberately neglect technical details. Moreover, because of lack of space, the treatment of the subject is very succinct. An extended version of this chapter can be found in [2].

Planning is a decision making activity regarding the actions to be taken to reach a goal. Traditionally, planning in AI focuses on autonomous agents, such as robots and unmanned vehicles. Such agents must be able to act within an environment in an autonomous way, i.e., without human intervention. In order to reach such an aim, an agent must modify the environment to achieve the desired situation or goal. Therefore, it must be provided with a representation of the environment where it operates and must be capable to reason in order to devise a sequence of actions enabling it to reach its goal. Planning has a long tradition in AI. The most influential approach in planning is surely that of STRIPS (Stanford Research Institute Problem Solver), the first major planning system, developed by Richard Fikes and Nils Nilsson in the early 1970s [3]. The language proposed for STRIPS still provides the basis for most languages used to date for expressing planning problems. Key terms in planning are *world*, *action*, *goal*, and – obviously – *plan*. The *world* is the environment which we take into consideration: it is composed by all the entities that affect our system. A *world state* is a "snapshot" of the world, i.e., the description of the entities and the properties of the world at a particular moment. An *action* produces a *state transition*, i.e., turns a world state into another world state by changing some entities or some properties of the world. A *goal* is a world state that we desire to reach. A *plan* is a sequence of actions required to bring the world into a goal state starting from a given initial state. Note that we consider first *classical* planning, that is, we assume that the world is fully observable (i.e., everything that happens in the world is known to the system), deterministic (i.e., an action performed in a state uniquely determines the resulting state; actions cannot fail during execution and always produce the expected effects), and static (the world cannot change due to events not in control of the system). Then, we briefly describe some approaches addressing scenarios in which some of these assumptions do not hold.

More formally, in classical planning, actions and the world are described by means of propositional logic. A classical planning problem can be characterized by a set of state variables, the initial state, the schemata of actions, and the goal states. More specifically, the parameters that define a planning problem are:

- a finite set of *state variables*, i.e., Boolean propositions that characterize a state of the world. State variables represent the features by means of which the world is described. They are literals in propositional logic;
- the *initial state*, described using the state variables;
- a finite set of the possible *actions* that can be performed. An action is a function that transforms a state of the world into another state. The states can be succinctly described by means of propositions over state variables. Therefore, actions can be described in terms of two components:

  - the *precondition*, that is a logical formula over state variables, which specifies the set of states in which the action can be performed; and
  - the *effect*, which describes the resulting states after the execution of the action in terms of the changes imposed on the value of the state variables;
- a finite set of *goal states*, represented by means of a propositional formula over the state variables.

A plan is a sequence of actions. The solution to a classical planning problem is a plan which transforms the initial state into a goal state.

1.1. Planning with State-Space Search

In order to solve a planning problem, one could devise a simple-minded algorithm: (i) systematically generate a transition graph with all possible states given by all possible sequences of actions, and (ii) find a path starting from the initial state and ending in a goal state. Since a path corresponds to a sequence of actions, such a path found by the algorithm corresponds to a plan. The problem of such an approach lies in its inefficiency: with as little as 30 state variables, there are $2^{30} = 1{,}073{,}741{,}824 \approx 10^9$ possible states (in fact, each variable can be evaluated to be either true or false). Therefore, since such a simple-minded approach is unfeasible in practice, it is necessary to devise an approach which, for example, does not explicitly generate all states beforehand. Actually, because of the combinatorial nature of most planning problems, especially when dealing with real-world scenarios, a great amount of work in the planning area has been devoted to improve the efficiency of planning algorithms. Details of planning with state-space search and efficiency-enhancement techniques are sketched in the following.

In ***progression planning***, the search for a plan starts from the initial state. At each step, the new successor state is generated. Progression means computing the successor state of a (current) state with regard to an action. By considering the precondition component of the actions, it is possible to determine which actions are applicable in the current state: they are the actions whose precondition formula is satisfied in the current state. By considering the effect component of the actions, it is possible to determine the new states resulting from the application of the actions by modifying the current state according to the effects: the successor state is obtained by adding the positive literals and by deleting the negated literals. As soon as a goal state is found, the search process stops: the solution is the sequence of actions applied in order to reach the goal state starting from the initial state. Progression planning can employ classical search algorithms, such as depth-first search or breadth-first search.

In ***regression planning***, differently from what happens with progression planning, the search for a plan starts from the set of goal states. The starting point is the formula which describes the goal states. At each step, given a formula representing a set of states, an action is selected and the *regression* operation is applied. Regression consists of computing a formula representing the predecessor states of the states represented by another formula. Considering the precondition/effect components of the action, the predecessor state is obtained by deleting the positive effects from the current state and adding the preconditions. As soon as a set of states is found such that it contains the initial state, the search process stops. While it is possible to select any action, for the sake of efficiency, it is convenient to guide the backward search by considering only those actions which actively contribute in reaching the initial state. Taking into account their effect component, these are the actions which achieve some literals in the current state and which do not negate any desired literals. Since, at each step, sets of states are considered, regression planning is potentially more efficient (but more complicated to implement) than progression planning, which considers only one state at a time. Classical search algorithms can be employed also when dealing for regression planning.

Unfortunately, both progression planning and regression planning perform poorly in domains involving a large number of state variables and actions. Therefore, it is necessary to increase the efficiency of the search process by avoiding a systematic search of the entire search space. This is done by exploiting problem-specific knowledge by means of ***heuristics***. In fact, the role of heuristics is that of guiding the search process, focusing on those actions which seem more “promising” (compared to the others). For example, a heuristics can be an estimate of the distance between the current state and the goal states, and focus on actions that decrease such a distance. An example of distance is the number of actions to be performed to reach a goal state from the current state in progression planning. Thanks to heuristics, planners can handle real-sized problems. The computation of the exact distance is actually impractical, because such a task is as hard as planning itself. Therefore, only estimates (i.e., approximations) are feasible. Such heuristics are employed with some well-known search algorithms exploiting heuristics estimates, for example A* and IDA*. Most heuristics in search-based planners are based on a particular data structure called planning graph [4], which proved to be very effective. A planning graph is an approximation of the transition graph and it has some useful properties: specifically, it can be constructed very efficiently and it enables to accurately estimate the cost of planning from a source state to a destination state.

1.2. Partial-Order Planning

Both progression and regression planning generate totally ordered plans, i.e., plans where actions are linearly ordered. On the other hand, a partial-order planner [5] does not commit in choosing the exact sequence of every action, but it rather “underspecifies” the generated plan, leaving some actions partially ordered, i.e., it does not specify for each pair of actions which must be executed first. There are some advantages in partial-order planning: partial-order plans are more flexible and it is easier to adapt them to some possible execution failure; such flexibility is useful also when combining smaller plans into larger plans, because possible conflicts can be avoided by reordering some actions. Partial-order planning is based on the *least-commitment principle*, i.e., never make a choice unless required to do so. In other words, a planner must make a choice only if it is relevant to solving the current part of the problem. For example, if two actions must not necessarily be sequential, they must be left unordered. While in progression and regression planning the search is performed in the space of states, partial-order planning can be implemented as a search in the space of incomplete plans. In this space, a node is an incomplete plan; the starting node is the empty plan (usually represented with two dummy actions, Start and Finish). An arc is a refinement of an incomplete plan. The final node is a complete partial-order plan which represents a solution of the planning problem. While in state-space search incomplete plans are extended by adding actions and generating new states, in plan-space search, incomplete plans are extended, for example, by adding a new step to the plan and by ordering two actions. An incomplete plan can be characterized by its open preconditions, i.e., those preconditions (or goals) which are not achieved by any action in the plan. Planners consider the open preconditions and try to achieve them by refining the incomplete plan, until there are no more open preconditions. The refined plan must avoid threats, i.e., a situation where an action, when executed at the wrong time, undoes some preconditions needed by another action. More precisely, a partial-order planner (i) chooses an open precondition, (ii) generates a refined plan by adding to the plan an action that achieves the open precondition, and (iii) it resolves threats by adding ordering constraints to the plan. If the set of open preconditions is empty, then the planner has found a complete partial-order plan which is a solution of the original planning problem.

1.3. Planning as Satisfiability

Because of recent advances in propositional-formulae satisfiability testing programs (SAT solvers), researchers took interest in studying the reduction of planning problems to propositional satisfiability problems (SAT) [6]. A SAT solver is provided with a propositional formula (i.e., a formula consisting of two-value variables – true and false – related through the operator of negation, disjunction, and conjunction). Then the SAT solver determines whether such a formula is satisfiable, and which assignments to the variables make it satisfied. The basic idea of planning as satisfiability consists in using SAT solvers instead of specialized algorithms for solving planning problems. Each formula encodes an entire plan; therefore it contains propositions that represent the initial state, the goal states, and all the executable actions, with their preconditions and effects. According to the truth values assigned to propositions corresponding to the possible actions, it is possible to infer which actions are part of a plan. There are many feasible ways to encode a plan in a propositional formula, different syntactically, but equivalent from the point of view of their "semantics". Some encodings are more compact than others with regard to the number of variables and/or to the size of the formulae; moreover different encodings may result in quite distinct computational profiles. An encoding which has proved to be efficient is based on planning graphs [7] (see Section 1.1). A main disadvantage of SAT planners lies in the encoding size; in fact, the number of variables and the size of the formulae can be very large and require huge amounts of memory. Hence, the importance of having compact encodings.

1.4. Hierarchical Task Networks

Hierarchical Task Networks (HTN) [8] are based on the idea of decomposing a planning problem into a number of smaller (and simpler) planning subproblems. HTN planning can be viewed as top-down planning. It captures the hierarchical structure of a domain: the basic idea is to start from an abstract plan and to reduce it to a more concrete plan, by expanding progressively each abstract action into a more concrete plan. HTN planning is useful when a domain expert can identify standard operating procedures to solve parts of a planning problem. For example, an abstract action such as Multiple Myeloma Treatment can be reduced to two more concrete but still abstract actions: Prednisone Treatment and Melphalan Treatment. The two treatments can – in their turns – be refined into more concrete actions, and so on, until there is an actually executable plan. Methods define how to reduce an abstract action into more concrete actions (a "task network") which may have some preconditions. Actually, preconditions and effects can include also logical inferences, numeric computations, and interactions with software packages. Task networks may involve also constraints, restricting, for example, the order in which the actions are to be performed (see Section 1.5.1).

A solution of the planning problem is generated by recursively applying methods in order to replace abstract actions with actions which are directly executable.

The definition of HTN is a way to encode user intent, i.e., capture expert advice about how to derive desirable solutions. In fact methods can be seen as “recipes”, i.e., standard ways about how to implement abstract plans. Therefore, differently from what happens in classical planning, in HTN planning, plans are not generated from first principles every time the planning problem has to be solved.

HTN planners provide several benefits compared with classical planning approaches: specifically, they are more expressive and allow to represent some problems which could not be represented with classical planning [9]; moreover, they are usually much more efficient. This is the reason why many planning systems used in real-world domains rely on HTN planning. The disadvantages of HTN planners lie in the fact that their definition actually embodies search control knowledge about how to derive a solution. As a consequence, it is much more complicated writing (and maintaining) a knowledge base than writing (and maintaining) classical planning operators. Moreover, they are less flexible, because they cannot handle problems that are not explicitly anticipated by domain experts.

1.5. Extensions to Classical Planning

Research in classical planning achieved thorough results. At the beginning it involved essentially toy problems (such as blocks world problems), which are quite different from real-world problems. Then, research has evolved along two lines.

The first line of research – active since the beginning and ongoing today – consists in dealing with real-sized problems, involving huge numbers of variables and states, striving for gaining more efficiency in solving planning problems. As briefly discussed in the previous section, in the last few years there has been a significant scale-up in the size of the plans that can be synthesized by planners, thanks to both new heuristics based on planning graphs and reduction to SAT problems.

The second line consists in realizing that the assumptions on which classical planning relies seldom match with real-world problems (and with CG management problems in particular – see also Section 2). In classical planning it is assumed that: (i) the actions are instantaneous, or, in general, have the same duration; (ii) resources are described completely by Boolean values (while in real world they can have also integer or real values); (iii) the environment is completely *static*: the planner can synthesize a plan beforehand and then execute it “blindly”, because the world does not evolve independently from the actions of the planner; as a consequence, the planner does not have to “sense” the environment; (iv) the actions are *deterministic*: if an action is performed in a state, it surely results in a specific state; (v) the environment is *completely observable*: every state of the environment is completely known.

Since this second line is more recent, there is not yet a comprehensive approach that can deal with planning problems by releasing all the preceding assumptions, but there is a wide variety of approaches addressing only some issues. Since it is not possible to cover every significant approach here, only the basic ideas related to such approaches will be mentioned.

1.5.1. Planning with Time and Resource Constraints

In the previous sections we have described some techniques to solve classical planning problems. As stated above, in classical planning we consider what actions are to be executed, and not when they will be executed or how long their execution will take. In other words, in classical planning it is assumed that actions are instantaneous and always executable, disregarding time. Moreover, we do not consider the case where the actions require resources (e.g., money, equipment, technicians) to be executed and how these resources have to be allocated. Traditionally, these two aspects of problem solving have been treated as two separate issues, the first one (what actions to be executed) in the area of planning, the last one (given a set of actions, when to execute each of them in order to obey the temporal and resource restrictions) in the area of scheduling. Moreover, scheduling approaches usually consider cost functions, in order to have an optimal scheduling as regards, e.g., total execution time and resource consumption.

One could solve a problem with time and resource constraints in two phases: in the first phase, s/he solves a planning problem, determining a set of possibly partially ordered actions that meet the goals of the problem; in the second phase, s/he solves a scheduling problem in which temporal and resource constraints are added to the solution of the planning problem. However, this approach is not much satisfactory since in real world it is possible to have problems in which there are complex interactions between the choice of the actions and the temporal and resource constraints. For these reasons, in the last years the distinction between planning and scheduling is gradually fading and research in planning and scheduling areas has started to merge, in order to combine the efforts of the two communities for solving such problems in an integrated way. A complete account of scheduling techniques and of integration of planning and scheduling is outside the scope of this section. Moreover, this is an active area for further research in order to fully integrate the results obtained separately in the two areas. Therefore, here we limit ourselves to shortly present a way of approaching the problem, based on constraint satisfaction techniques, successfully exploited for scheduling problems.

A Constraint Satisfaction Problem (CSP) [10] is defined by a finite set of variables – where each variable is associated with a domain – and a set of constraints over a subset of variables, restricting the possible combinations of values that the participating variables may assume. A solution of a CSP is obtained by assigning to each variable a value that is consistent with the constraints. Such framework can express complex temporal and resource constraints and supports specialized reasoning schemes, for example for the management of temporal constraint networks (STP and DTP [11]).

There is a wide variety of approaches in planning and CSP with different degrees of integration between the conventional planning component and the CSP component. For example, Constraint-Based Interval (CBI) planners [12] represent actions and propositions as intervals and they rely on constraint-based techniques to manage the relations between the intervals. Such planners are coupled with a CSP component that can act as an add-on, for example, to check whether numerical relations such as temporal and resource constraints are satisfied. The main drawback of such approaches lies in a lack of efficiency, because they do not scale up as well as state-of-the-art classical planning approaches.

HTN planning is particularly suitable for planning with constraints. In fact – as we have seen in Section 1.4 – it is possible to define specific constraints for planning subproblems at each level of decomposition (with the constraints associated with task networks). Such constraints may impose a wide range of restrictions, such as temporal, resource, cost restrictions etc. The decomposition derived from the hierarchically structured description of the planning problem allows to efficiently propagate the constraints.

1.5.2. Planning with Uncertainty, Non-determinism, and Incomplete Information

In real-world settings, it is not possible to know a priori the result of executing an action. For example, the action may fail during execution, or the agent may have incomplete information about the current state which does not allow to foresee what the exact outcome of the action will be. We consider separately the case of *bounded indeterminacy*, where, for example, it is not possible to anticipate the exact outcome of an action, but it is possible to list all the possible outcomes, and the case of *unbounded indeterminacy*, where there can be completely unanticipated outcomes.

Regarding ***bounded indeterminacy***, two different approaches have been proposed to deal with these problems: *non-deterministic planning* and *probabilistic planning*. Due to lack of space, we focus on probabilistic planning only, which is closer to some CG approaches. For a more complete account, refer to [2].

In probabilistic approaches, uncertainty is dealt with by means of probabilistic models. While in non-deterministic approaches an action may have more than one outcome without "preferences" between them, in probabilistic approaches the outcomes of an action have an associated probability distribution, i.e., the possible outcomes are labeled with their respective probabilities.

The probabilistic frameworks are based on Markov Decision Processes (MDPs) and their extensions [13]. The solution of an MDP problem is not an actual plan, but a policy, which contains a mapping from any state to the action that should be taken in that state. Partially Observable Markov Decision Processes (POMDPs) [14] are an extension of MDPs which can deal with the case of partial observability.

The MDP framework is highly expressive in handling stochastic dynamics and uncertainty, and it is possible to specify a variety of objective functions and optimization goals. However, it has some major drawbacks. In fact, it provides an inadequate representation of time (it is usually assumed to be atomic), and it is highly expensive from a computational point of view: it is prohibitive to represent real-sized planning problems. Moreover, such frameworks need accurate estimates of distribution probabilities, which are difficult to acquire. However, recently some smooth transitions between pure non-deterministic approaches and probabilistic approaches are being investigated, in order to unify both approaches in a unique framework with imprecise or qualitative estimates (see, e.g., [15]).

Unbounded indeterminacy is a more realistic assumption for CG application: actions can fail in unexpected ways, and the goals or even the environment itself may change during the execution of the plan. Moreover, it may be infeasible, or costly, to explicitly represent such great amount of information. In this setting, it is not possible to build into a plan a branch for every possible contingency, and it is necessary to perform some form of planning during execution, when some unexpected event occurs.

With unbounded indeterminacy, plan execution requires more than blind adherence to the previously generated plan, and the agent should be monitoring the environment and the actual effects of its actions [16]. When an unanticipated event is detected, the agent has to make a decision at runtime in order to formulate a new plan (replanning) or to adapt the current plan (repairing). In devising the new plan, the agent may have to be conservative, because, for example, the part of the plan already executed has made some commitments with some external agents and they cannot be disrespected. As examples of unanticipated events there may be failures in the execution of some actions and changes in the environment state so that some needed information is no longer available or the preconditions of the remaining steps in the plan are no longer valid.

This line of research has not been explored in depth to date. As far as we know, it includes few applicative works and even fewer methodological works.

1.6. Beyond Planning in Artificial Intelligence

Traditionally, the planning community has almost exclusively stressed the aspect of planning related to the automatic generation of plans from the domain theory and the problem specification. We have provided a rapid survey on the general ideas at the basis of planning in AI. We started with describing the main approaches in classical planning, which exhibits well-established results but it relies on many limiting assumptions, which prevent it from being directly applicable to the domain of CGs. Then we moved to describe some extensions to the classical planning paradigm that relax some assumptions. Among them, we can identify various approaches which are already being applied in some systems for the computerized management of CGs. In particular, *CSP* is being resorted to for (i) managing temporal information and reasoning with it (see Section 2.2); (ii) managing non-temporal resource information (see Section 2.1); *probabilistic planning* is being exploited for providing decision support to physicians, in particular as regards therapy selection (see Section 2.2).

However, the research in this area is ongoing and several issues still need to be dealt with. The research agenda (see Section 4) analyses this aspect in further detail.

2. Planning in the Clinical Guidelines Context

Differently from planning in non-medical contexts, planning in medicine, and in the CG management domain in particular, does not require to edit a general procedure from scratch, since such a procedure typically already exists. Actually, CGs are exactly a set of suggested actions, to be followed in a suggested order, provided by a committee of experts. Nevertheless, the form in which CGs are issued may often be imprecise or not completely constrained, or may include different uncertainty elements. As a matter of fact, CGs can be seen as *skeletal plans* [17], i.e., schemes which capture the main lines of a procedure, but leave space for execution-time adjustments and selections among alternatives, both in the actions to be completed, and in the actions flow.

In order to better examine the characteristics of planning in the CG management domain, we first need to clarify what we mean for *state*, *action*, and *state transition* (see Section 1) [18] in the process modeled by the CG itself.

It is straightforward to define the concept of *state* as the set of patient's parameters that are measured for taking decisions and for assessing therapy outcomes. Each parameter is a *state variable*. Clearly, state variables are not necessarily Boolean. Moreover, some required state variables might be missing in practice.

State *transitions* are changes in the patient's state variables, due to the effect of an *action*. In particular, we can consider the state transition between time t and time $t+1$ as due to all the *work actions* (i.e., the medical procedures actually executed on the patient) between the two time instants. Actually, this is a simplification, since patient's parameters can vary due to exogenous reasons (by exogenous, we mean "not due to any action described in the CG"). Typically, some state variables can change because the patient becomes older, or because s/he catches another disease. To explicitly represent all these possibilities, one should be provided with a whole model of the patient's behavior and of all the stochastic variables that could influence it. This kind of information is normally not explicitly available in clinical practice. However, the probability that a set of actions (typically implementing a therapy) produces a certain effect on a patient (i.e., the probability of a certain state transition) is a number introduced in the CG as provided by the medical literature: this number is obtained by drawing statistics on real cases, and real cases cannot separate the effect of the therapy from the exogenous effect.

The same *action* might have a different duration, or uncertain effects when applied to different patients, due to the single persons' peculiar characteristics. In case of problems, actions can also be suspended or substituted with others.

Given these clarifications, we can now summarize the most relevant characteristics of planning in the CG management domain - see also [19]:

- information about the patient state may be incomplete;
- state variables could change due to exogenous reasons, whose effect cannot always be easily distinguished from the one of intended (therapeutic) actions;
- effects of actions on the patient states may be non-deterministic;
- an action duration might change, due to various (and maybe unpredictable) reasons;
- actions might have vanishing or delayed effects;
- plans might need to be suspended.

These features, along with the observation that a CG can be considered just as a skeletal plan, render the management of uncertainty and of temporal issues very relevant in our domain, and lead us to say that planning in the CG context can be more properly interpreted as a problem of optimized *execution*: actually, a significant support to CG execution, and to CG adaptation to environment and patient peculiarities, needs to be provided to physicians. More specifically, four main tasks need to be accomplished at execution time [19]:

1. *plan modification* (in order to change/adapt a skeletal plan to the actual and current patient and hospital local needs);
2. *plan execution* (in order to perform the –possibly modified – activity flow);
3. *plan critiquing* (in order to analyze the CG as it was executed in the past, with the aim of ameliorating performances in its future adoptions);
4. *plan visualization* (in order to provide proper representation and communication facilities to users who execute the CG in clinical practice).

In the rest of the chapter, we will therefore analyze the recent contributions by the main groups of researchers working on systems for the computerized management of CGs, according with this categorization. Links to other chapters of the book, treating in greater detail some of the cited aspects, will be provided where appropriate.

2.1. Plan Modification

Plan modification probably represents the largest and most interesting task among the ones listed in Section 2; as a matter of fact, in their conventional, paper-based form, CGs often present population-oriented recommendations, and are usually far from being point-of-care facilities [20]. Several approaches to deal with the issue of CG change and tuning, involving different methodologies, have thus been proposed in the literature. Schematically, they can be subdivided as follows: (i) approaches that deal with CG adaptation; (ii) approaches that deal with non-compliance management.

As regards **CG adaptation**, it has been observed that CG dissemination and integration into clinical practice should recognize the multiplicity of working settings and information system environments, within which the CGs themselves are meant to be implemented [21]. In particular, CGs management systems, to make CGs really usable in practice, need to deal with two types of adaptation: (i) local adaptation, i.e., adaptation to local constraints (see also [22]) in local settings (e.g., hospital resources availability, available practitioners' skills), and (ii) cultural adaptation, i.e., the adaptation related to the fact that different countries and/or cultural settings may have different degrees of acceptance of specific clinical procedures, and/or local best practices. Moreover, a closely related problem is what we call (iii) upgrade adaptation, i.e., the fact that, periodically, CGs have to be updated in order to include relevant novelties in the clinical field (e.g., new therapies) and, possibly, to remove obsolete choices. Different computer-based approaches to local adaptation have been proposed in the literature. [23] proposes to extend the CG representation formalism to take into account the resource requirements associated with each action (and, therefore, to each alternative path in the CG): only those actions (paths) whose resource requirements can be satisfied in the given context (e.g., hospital) can be executed. The approach provides an algorithm that takes as an input a "general" CG, and a list of all the locally available resources, and gives as an output a new CG, in which all the paths containing non locally executable actions have been pruned away. [24] deals with the double problem of local adaptation and of integration with the patient's electronic medical record, and tests the approach on a real-world CG. The authors observe that the two issues at hand have significant affects on the encoding of the CGs, including change of algorithm design, definition of decision criteria, and specification of data items that are referenced by the decision criteria themselves. Therefore, CG adaptation and integration with the medical record should be considered as early as possible. Again as concerns the adaptation based on resources availability, one abstract solution which has been proposed is to have a high-level description of the CG *intentions*, in order to ensure the adaptability of the procedure to different contexts still preserving the CG intentional objectives [25]. Such an approach has been followed in CAMINO [26], a tool that provides a user-friendly interface to modify (e.g., by adding/removing/changing actions) a CG, using additional information about the hospital. [21] suggested an approach in which the dependencies between actions in a CG can be explicitly described, and where users' modifications to a general CG must respect these dependencies.

The above solutions provide facilities to help physicians modifying CGs (consistently with the CG intentions and/or functional dependencies), but basically deal just with local adaptation. A more recent contribution [27], on the other hand, focuses on an integrated treatment of cultural and upgrade adaptation, by supporting cooperative work in the domain of CGs' definition and maintenance, by (i) distinguishing between different levels of authors and by (ii) maintaining the history of the updates of the CGs. To this end, the authors introduce a three-layered architecture, in which the cooperative work facility, available through an interface layer, is based on a query language and on a data model layer. Such an approach properly extends classical Temporal Databases semantics in order to cope with the issues introduced by cooperative work. This methodological contribution will soon be implemented within the GLARE system [28]. The interested reader will find additional aspects on CGs adaptation in Chapter 7.

The second issue related to the plan modification task is **non-compliance management**. Despite the efforts to facilitate CGs' adoption and integration in real-world environments, provided by the tools for computerized management of CGs, and more specifically by the adaptation facilities described in the previous section, physicians may decide to not follow the CG indications. Basically, these non-compliance episodes may emerge due to two categories of reasons: 1) the physician has to face a somehow unpredicted situation, which was not (properly) considered when the CG was issued; 2) her or his professional opinion is different from the one expressed in the CG, and s/he does not completely trust the CG itself. Keeping track of such deviations from the default CG execution, and documenting the physician's motivations, is clearly an added value. Moreover, repeated alterations of CG tasks (or flow) may indicate an improper or weak initial CG definition, and might be used as a starting point for suggesting a formal CG revision to a committee of expert physicians. More details about the non-compliance problem definition can be found in Chapter 9. Non-compliance with CGs are treated in GUIDE [29]. The system allows a user to redirect, delay, or be non-compliant with a CG task. Non-compliance may lead to the execution of a different task, still chosen among the ones described in the CG (e.g., by altering the control flow), or to the definition of a new task: in this case, the operator has to select it from the SNOMED taxonomy. The requirement is that the *intention* (see above) of the CG is still reached, even if by means of a different procedure. Intentions allow to verify whether the physician is still compliant with the CG at a high level (i.e., s/he still shares the CG goals), even when an exception is generated at the low level (because s/he has changed some tasks or execution flows). A formal description of intentions is provided in EON [30], PRO*forma* [31], and Asbru [25,32,33]. In Asbru, in particular, any CG modification is released only after its high-level compliance with the CG intentions has been verified by means of a rule-based system. The Asbru verification facility also allows to discover anomalies (such as non-satisfiable conditions) that were originally introduced in the CG during the acquisition phase. Other works address the issue of CG verification, e.g., by means of model checking [34,35] or theorem proving [36] techniques. However, the principal aim of these works is to discover logical inconsistencies in the CG or to prove particular properties it exhibits, while in this section we are basically interested in keeping track of non-compliances (typically to a well-formed CG) due to reasons of type 1 and 2 presented above. Chapter 4 of this book treats verification issues. Finally, the work in [37] proposes an approach for managing non-compliance with CGs, based on the Case-based Reasoning (CBR) methodology. In front of a new non-compliance case, the tool allows the physician to retrieve past situations similar to the current one, and to decide whether to re-apply the same CG modifications adopted in them. Moreover, the tool is able to learn indications from the non-compliance cases, that can be deployed to suggest CG revisions. This CBR-based management of non-compliance is more "lazy" with respect to the other discussed literature approaches, since it does not aim at checking on-the-fly the adherence of a modification to the original CG intentions; thus, it does not model goals or verification rules, whose elicitation might be extremely hard and time consuming. It simply supports physicians' revisions by showing past non-formalized examples that match the current context, and leaves her/him the responsibility of the final decision. Keeping track of such cases requires a very limited knowledge acquisition effort (since just the instances of CGs as they were edited and executed in the past have to be stored), and small memory requirements (since non-compliances are expected to be infrequent). Then, when some cases related to the same context have been acquired, the approach tries to learn some more general suggestions from them, thus preparing a structured CG revision proposal supported by a set of concrete implementations. In the literature, CBR has often been resorted to in order to maintain knowledge about exceptional situations (see, e.g., [38,39] as applications in the medical domain). In particular, in [38], a CBR approach has been devised to support the physician in adapting a specific therapeutic protocol to a usual situation (we could say: in defining a non-compliant version of the protocol; the protocols at hand can be seen as skeletal plans). Nevertheless, only temporary variations of the protocol, for a particular patient, are considered. Long-term protocol changes, obtained from frequently performed adaptations, are not dealt with. In particular, the issue of reorganizing the case base in order to learn well-defined suggestions for supporting CG revision, which is a significant need [40] in practice, appears to be a relevant and original contribution of the approach in [37].

2.2. Plan Execution

After a possible modification phase, carried out along the lines described in the previous section, CGs need to be executed, i.e., applied to a real patient in a real environment. In order to support physicians in this task, a large number of the computer-based systems for CGs management proposed in the literature have integrated “execution” facilities [20]. However, existing execution engines are mainly proprietary developments by the authors of the models, showing peculiar choices and a somehow limited generalization capability. A few exceptions to this statement exist, represented by systems in which: (i) the authors have made the effort of operatively integrating the CG process with the organizational environment characteristics and needs, basically taking advantage of workflow management systems concepts; (ii) the authors have enhanced the system functionality by means of decision support methodologies, basically meant to help physicians in taking non-trivial (therapeutic) decisions (i.e., decisions in which no alternative is actually better than the others, from a strictly clinical viewpoint).

Workflow management systems integrate domain and organizational knowledge to support business processes. When applied to the medical environment, and in the domain of CG implementation in particular, they may be used to manage care delivery by enhancing co-operation among healthcare professionals. The Workflow Management Coalition [41] model has constructs for expressing nesting, iterations, branch selection, and some temporal constraints (e.g., synchronization). Therefore, it seems to be a potential common control-flow model for CG management systems. As a matter of fact, several authors are working at mapping CG execution primitives to the workflow model, and at exploiting the workflow technology for **integrating CGs with the organizational environment** in which they are meant to be applied. The system GUIDE [29], which has been built on top of enterprise workflow standards and tools, is probably the most representative system in this sense. In particular, the Patient Workflow Management System developed in GUIDE is based on a detailed model of both the medical work process (i.e., the CG) and the organizational structure. The authors have also developed a mechanism for translating the CGs into a computational formalism, precisely a Petri Net, which takes into account the specific organization characteristics and allows allocating resources for managing a specific patient in daily practice [42]. Also Peleg's group has started working on the topic; they have mapped instances of the GLIF ontology to the reference model of the Workflow Management Coalition using Protégé ontology mapping rules [41]. This model can be mapped further on Petri Nets for verification of structural properties and for studying the system behavior. However, despite the fact that most of the written CGs implicitly define a workflow process, healthcare organizations are very different from industrial or commercial companies. Their main goal is not profit, but maintaining and improving the public health: therefore, outcomes are difficult to measure; moreover, physicians are quite independent decision-makers; finally, patients themselves may be involved in choosing treatment options. For these reasons, the standard functionality of typical workflow management systems needs to be strongly enhanced in order to cope with healthcare delivery needs, and the workflow standards need to be significantly extended to represent all CG primitives. It is still unclear if such extensions can all be mapped to Petri Nets as well: such mapping, if possible, should support formal verification of CG properties.

It is worth mentioning that the authors of GLIF have also investigated an approach of Guideline Execution by Semantic Decomposition of Representation (GESDOR) [43]. By properly mapping primitive tasks - such as data collection, decision making, branching, and synchronization as well as auxiliary tasks such as criterion evaluation and event management - GLIF and PRO*forma* models have been successfully executed by the GESDOR engine. Ultimately, GESDOR might become a universal execution engine for CG models [43]. However, while GESDOR is a generalization from concrete CG execution engines, it is still not based on or compatible with industrial workflow engines and the question whether these generalized tasks can be mapped to workflow standards like XPDL or BPEL remains open.

As regards **decision support in CG execution**, several techniques have been proposed in the literature to this end. Some of them are rather simple approaches, in which, e.g., a first-order language is adopted to represent rules aimed at selecting among (therapeutic) alternatives in the CG [20]. More complex approaches may be distinguished in two main categories: (i) *probabilistic frameworks*, and (ii) *temporal frameworks*. Among the approaches in category (i), we need to cite the GLARE system [18,44], which embeds decision theory concepts to support therapy selection. The authors start from the observation that in several situations no alternative is really "better" than the others, from a strictly clinical viewpoint, and CGs, being skeletal plans, are only meant to present all the range of choices, leaving to the user the responsibility of selecting the "right" one. Even when resorting to a computer-based system for CGs management, just "local" information, describing the decision at hand, are normally shown to the user. On the other hand, the possibility of obtaining a complete scenario of the decision consequences (in terms of the probability of the different therapy outcomes, of therapy utilities, and of money, time, and resources spent following the different paths) would be clearly an added value for physicians. In clinical practice, various selection parameters (such as the costs and effectiveness of the different procedures) are sometimes available when executing a CG, but the task of comparing and balancing them is typically left to the physician. Decision theory seems a natural candidate as a methodology for affording this analysis: the authors thus propose a mapping between CG primitives and decision theory concepts, on which they are building the implementation of a tool able to automate the comparison of different alternatives and to provide quantitative results. Still in category (i), we find the contribution of GUIDE [45]. As anticipated above, in GUIDE, before being executed, a CG is translated into a high-level Petri Net, a rigorous formalism for modeling concurrent processes. The resources, necessary for performing CG-based activities, are also represented by means of an organizational model. This allows the execution of the Petri Net for simulating the implementation of the CGs in the clinical setting. The purpose of the simulation is to validate the model and to suggest the optimal resource allocation before the system is installed (see also [41]). Moreover, in GUIDE the workflow management technology is applied to support CGs execution; the requirements of the domain have imposed to enhance workflow management systems by ensuring the flexibility and the uncertainty management typical of the health-care processes: in particular, the authors have chosen to resort to decision trees and influence diagrams in order to select among alternatives, in the case of non-trivial choices, as the ones illustrated in the case of GLARE. Approaches in category (ii), on the other hand, offer support for representing and/or reasoning with temporal information. Temporal information can consist in constraints about duration of actions, delays between actions and periodic repetitions of actions. In this category, several works should be cited. GLIF [46,47] deals both with temporal constraints on patient data elements and with duration constraints on actions and decisions. In PRO*forma* [31], it is possible to define constraints on the accomplishment of tasks, as well as task duration and delays between tasks. Moreover, temporal constructs can also be used in order to specify the preconditions of actions. DILEMMA and its successor PRESTIGE [48] model temporal constraints within conditions. EON [30] uses temporal expressions to allow the scheduling of guideline steps, and deals with duration constraints about activities. Moreover, by incorporating the RESUME system, it provides a powerful approach to cope with temporal abstraction. In EON, the Arden Syntax allows the representation of delays between the triggering event and the activation of a Medical Logic Module (MDL), and between MDLs [49]. While the above-mentioned approaches stress the representation aspect of temporal information, other approaches deal also with the reasoning aspect, i.e., drawing inferences on the basis of temporal information. In [52], for instance, the goal of temporal reasoning is to find out proper temporal abstractions to data and properties. Interpolation-based techniques and knowledge-based reasoning are used. In Asbru, a comprehensive approach based on the notion of temporal constraint propagation has been proposed [32,50]. In particular, in [50], different types of temporal constraints – deriving from the scheduling constraints in the guideline, from the hierarchical decomposition of actions into their components and from the control-flow of actions in the guideline – are mapped onto a Simple Temporal Problem (STP) framework [11]. Temporal constraint propagation is used in order to (1) detect inconsistencies, and to (2) provide the minimal constraints between actions. There is also the claim that (3) such a method can be used by the guideline interpreter in order to assemble feasible time intervals for the execution of each guideline activity. Moreover, advanced visualization techniques are used in order to show users the results of temporal reasoning [53]. In [51] the authors provide a temporal representation formalism for temporal constraints including periodic constraints and reasoning mechanisms (performing inferences in the form of constraint propagation). They devise a Temporal Server loosely coupled with a CG system; they describe an implementation in GLARE, but the approach is system independent. The CG system delegates temporal-related problems to the Temporal Server. The authors show how the Temporal Server can be exploited in order to provide CG systems with different temporal facilities. Specifically, during the CG acquisition phase, it enables to represent temporal constraints and to check their consistency. During the execution phase, it allows the physician to check the consistency between action execution times and the constraints in the CGs, and to provide query-answering and temporal simulation facilities (e.g., when choosing among alternative paths in a CG). A good general introduction to temporal reasoning in medicine can be found in [54]. The topic of temporal abstraction is further investigated in Chapter 5.

2.3. Plan Critiquing

Expert critiquing systems were introduced since the 1980s, with the aim of supporting medical decision making, and operated by providing critique on a physician's decisions, rather than telling him/her exactly what to do [55,56]. The expert critiquing system paradigm can be fruitfully adopted in medical CGs as well, and allows an analysis of possible non-compliance episodes with the CG itself. Nevertheless, two major limitations with "classical" critiquing systems, which make them hardly acceptable in practice, can be recognized: (i) the fact that these systems do not cope with deviations from the underlying model, and (ii) the fact that these systems are not able to deal with the question why a physician was performing an action (which is clearly essential to provide a grounded critique) [57]. To address both limitations, [58] and [59,60] have recently suggested to perform critiquing by assessing the compliance of a physician's *intentions* with the intentions behind a CG. Asbru, EON, GUIDE, and PRO*forma* can support critiquing, since their languages formally specify intentions [20], with Asbru having the most extensive intention-modeling capacity [25,32,61] (see Section 2.1). In detail, a critiquing system must determine to what extent a CG has been followed by the physician in order to provide adequate critique. Hence, starting from a general high-level intention, the critiquing system must search through the possible execution of the CG, to see if it was correctly matched. In [56], in particular, a framework for intention-based matching of physicians' actions is reported. The distance between the prescribed CG actions and the practically implemented ones is calculated, after having generalized the CG details (e.g., by substituting specific drugs with the drug group to which they belong – like, for instance, quick response insulin). This approach is currently being tested within the Asbru project, in which the CG skeletal plan is defined, while the single sub-plans to be selected to reach the CG target are not. Instead, these plans are found basically by matching their intentions with those demanded by the skeletal plan. Similar critiquing efforts might be rather easily integrated in EON, GUIDE, and PRO*forma*; we also direct the reader to Section 2.1 for a more comprehensive treatment of non-compliances in the state-of-the-art literature.

2.4. Plan Visualization

Paper-based CGs are typically represented using flowcharts, decision tables, or plain text. These representations are badly suited for complex medical procedures. Formalized languages for representing CGs, such as the ones adopted in the CGs management systems described in the literature [20], are able to solve these limitations, but they incorporate many concepts from Computer Science, and are not very usable for physicians as well. Visualization enhancements and user-friendly graphical interfaces are therefore needed in practice. The most significant contribution in this direction is provided by the system Asbru. In Asbru, the authors use graphical metaphors to make the underlying concepts easier to grasp, employ glyphs to communicate complex temporal information, and colors to make it possible to understand the connection between the topological view and the temporal view available in the system [70]. Moreover, they are developing CareVis – an interactive visualization method to integrate and combine classical data visualization with the visualization of treatment information in terms of logic and temporal aspects [62]. Through CareVis, Asbru provides multiple simultaneous views to cover different aspects of a complex underlying data structure of treatment plans and patient data. These tightly coupled views exploit visualization methods well-known to domain experts and are designed to facilitate users' tasks. Complex effects borrowed from photography and cinematography are resorted to as well [63]. The user-centered development approach applied for these interactive visualization methods has been guided by user input gathered via a user study, design reviews, and prototype evaluations. For a more detailed description of facilities for CG visualization, we suggest the reader to refer to Chapter 8.

3. Conclusions

We have discussed the topic of planning from the different perspectives of AI and of CG management.

As regards planning in AI, we have described classical planning approaches and some heuristic techniques. Moreover, we have highlighted the differences between the area of planning in AI and that of planning in CG management, most of which are due to the specific features of the field of medicine. We pointed out the main requirements of a planning system to be profitably employed in CGs systems and we have observed how some recent extensions to the classical planning paradigm, addressing temporal and resource information support, as well as uncertainty and incomplete information support, can be (at least partially) resorted to, in order to support CG management needs. We concluded that, in the light of recent developments of planning in AI, the difference between planning in AI and planning in CG management is gradually becoming less prominent, even if a comprehensive planning system suitable for dealing with CGs has not been devised yet.

Regarding planning in the CGs context, it can be seen more as a support for an optimized CG execution. Many diverse formalisms and tools have been proposed to this end, with diverse features and various strong points. We have described the different approaches along four main tasks to be accomplished when executing. *Plan modification* regards adaptation of a skeletal plan to a specific patient and hospital. This issue has been addressed along two lines: (i) adaptation to local settings, such as hospital resources, adaptation to different countries and cultures and support for the evolution of CGs, and (ii) non-compliance management, addressing unpredicted situations and deviation from CG because of divergence of opinion among physicians. *Plan execution* in the proper sense also has been addressed along two lines: in fact, it has been regarded as (i) ensuring integration with organizational and information system environments, exploiting, e.g., workflow management systems, and (ii) providing decision support during the execution and performing inferences on the available data (about, e.g., resources and time) in order to support physicians to take fully informed choices when dealing with clinical actions to be implemented. *Plan* *critiquing* deals with the problem of assessing the compliance of a physician's intention with the intentions behind a CG, in order to provide critique on past physicians' decisions for improving the CG in future executions. *Plan visualization* addresses the issue of communication between computerized CG management systems and users who execute CGs.

Valuable results concerning planning in the CG context have thus been obtained; nevertheless, some open issues exist: an overview about them is presented as a research agenda for the future in Section 4.

4. Research agenda

As far as we know, there is not yet a comprehensive approach to planning in AI that deals with all the features needed in CGs, which require to support at the same time at least [19]: (1) resources, (2) costs, (3) temporal information, (4) partial observability, and (5) unbounded indeterminacy. Moreover, research in planning under-addressed other very important related topics, which are fundamental in dealing with CGs, as argued in [19] and in [64]: (6) plans have to be considered in a more general context, where the generation of the plan is only the beginning, and the whole plan lifecycle management has to be considered (see, e.g., [19]). In addition, (7) some level of interaction with users, such as mixed-initiative problem solving [64], where there is some form of collaboration between humans and computers, should be supported. In fact, planning systems would be not completely automatic but would assist humans, for example, in verifying plans and control resources, so humans could exploit their expertise and intuition in order to exercise some level of control. All the above-mentioned aspects (1-7) are open research issues for the future. Moreover, (8) the integration of CG systems both among themselves and with the organizational environment (e.g., workflow management systems) is another challenge for future investigation.

References

[1] D. Nau, M. Ghallab, P. Traverso, *Automated planning: theory and practice*, Morgan Kaufmann, San Francisco, CA, 2004.

[2] L. Anselma, S. Montani, Planning, Technical Report 101/07, Dipartimento di Informatica, Università di Torino, Italy (2007) (available at http://www.di.unito.it/~anselma/pdf/Planning TR101-07.pdf)

[3] R. Fikes, N. Nilsson, STRIPS: a new approach to the application of theorem proving to problem solving, *Artificial Intelligence* **2** (1971), 189-208.

[4] A. Blum, M. Furst, Fast planning through planning graph analysis, *Artificial Intelligence* **90** (1997), 281-300.

[5] D. Weld, An introduction to least-commitment planning, *Artificial Intelligence Magazine* (1994), 27-61.

[6] H. Kautz, B. Selman, Pushing the envelope: planning, propositional logic and stochastic search, in Proc. 13th National Conference on Artificial Intelligence (1996), 1194-1201.

[7] H. Kautz, B. Selman, Unifying SAT-based and graph-based planning, in Proc. International Joint Conference on Artificial intelligence (IJCAI'99) (1999), 318-325.

[8] K. Erol, J. Hendler, D. S. Nau, UMCP: A sound and complete procedure for hierarchical task-network planning, in Proc. International Conference on AI Planning Systems (AIPS) (1994), 249-254.

[9] K. Erol, J. Hendler, D. Nau, HTN Planning: Complexity and expressivity, in Proc. AAAI 1994 (1994), 1123-1128.

[10] R. Dechter, *Constraint processing*, Morgan Kaufmann Publishers, 2003.

[11] R. Dechter, I. Meiri, J. Pearl, Temporal constraint networks, *Artificial Intelligence,* **49** (1991), 61-95.

[12] D. E. Smith, J. Frank, A. K. Jónsson, Bridging the gap between planning and scheduling, *Knowledge Engineering Review* **15(1)** (2000), 47-83.
[13] C. Boutilier, T. Dean, S. Hanks, Decision-theoretic planning: structural assumptions and computational leverage, *Journal of Artificial Intelligence Research* **11** (1999), 1-94.
[14] L. P. Kaelbling, M. L. Littman, A. R. Cassandra., Planning and acting in partially observable stochastic domains, *Artificial Intelligence* **101(1-2)** (1998), 99-134.
[15] F. W. Trevizan, F. Gagliardi Cozman, L. Nunes de Barros, Planning under risk and knightian uncertainty, in Proc. IJCAI 2007 (2007), 2023-2028.
[16] K. L. Myers, CPEF: a continuous planning and execution framework, *AI Magazine*, **20(4)** (1999), 63-70.
[17] P.E. Friedland, Y. Iwasaki, The concept and implementation of skeletal plans, *Journal of Automated Reasoning* **1(2)** (1985), 161-208.
[18] S. Montani, P. Terenziani, Exploiting decision theory concepts within clinical guideline systems: towards a general approach, *International Journal of Intelligent Systems* **21** (2006), 585-599.
[19] S. Miksch, Plan management in the medical domain, *AI Communications* **12(4)** (1999), 209-235.
[20] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E.H. Shortliffe, M. Stefanelli, Comparing computer-interpretable guideline models: a case-study approach, *JAMIA* **10** (2004), 52–68.
[21] A.A. Boxwala, Applying axiomatic design methodology to create guidelines that are locally adaptable, in Proc. American Medical Informatics Annual Symposium (2002), 980.
[22] M.D. Cabana, C.S.N.R. Rand, C.S. Powe, A.W. Wu,. M.H. Wilson, , P.C. Abboud, H.R Rubin, Why don't physicians follow clinical practice guidelines? A framework for improvement, *JAMIA* **282(15)** (1999), 1458-1465.
[23] P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G. Molino, G. Correndo, A context-adaptable approach to clinical guidelines, in Proc. Medinfo (2004), 169-173.
[24] M. Peleg , D. Wang, A. Fodor, S. Keren, E. Karnieli, Adaptation of practice guidelines for clinical decision support: a case study of diabetic foot care, in Proc. Workshop on AI techniques in healthcare: evidence-based guidelines and protocols, ECAI (2006).
[25] Y. Shahar, S. Miksch, P. Johnson, An intention-based language for representing clinical guidelines, in Proc. American Medical Informatics Annual Symposium (1996), 592-596.
[26] D.B. Fridsma, J.H. Gennari, M.A. Musen, Making generic guidelines site-specific, in Proc. American Medical Informatics Annual Symposium (1996), 597-601.
[27] A. Bottrighi, L Anselma, S Montani, P. Terenziani, G. Molino, M. Torchio, Supporting cooperative updates of clinical guidelines, in Proc. Medinfo (2007).
[28] P. Terenziani, G. Molino, M. Torchio, A modular approach for representing and executing clinical guidelines, *Artificial Intelligence in Medicine* **23** (2001), 249-276.
[29] S. Quaglini, M. Stefanelli, G. Lanzola, V. Caporusso, S. Panzarasa, Flexible guideline-based patient careflow systems, *Artificial Inteligence in Medicine* **22(1)** (2001), 65–80.
[30] M.A. Musen, S.W. Tu, A.K. Das, Y. Shahar, Eon: a component-based approach to automation of protocol-directed therapy, *JAMIA* **6(3)** (1996), 367-388.
[31] J. Fox, N. Johns, A. Rahmanzadeh, R.Thomson, Disseminating medical knowledge: the proforma approach, *Artificial Intelligence in Medicine* **14(1-2)** (1998), 157-182.
[32] Y. Shahar, S. Miksch, P. Johnson, The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines, *Artificial Intelligence in Medicine* **14(1-2)** (1998), 29-51.
[33] G. Duftschmid, S. Miksch, Knowledge-based verification of clinical guidelines by detection of anomalies, *Artificial Intelligence in Medicine* **22** (2001), 23-41.
[34] S. Baumler, M. Balser, A. Dunets, W. Reif, J. Schmitt, A verification of medical guidelines by model checking -a case study, in Proc. Model Checking Software SPIN (2006).
[35] L. Giordano, P. Terenziani, A. Bottrighi, S. Montani, L. Donzella, Model checking for clinical guidelines: an agent-based approach, in Proc. American Medical Informatics Annual Symposium 2006
[36] A. tenTeije et al., Improving medical protocols by formal methods, *Artificial Intelligence in Medicine* **36** (2006), 193-209.
[37] S. Montani, Case-based Reasoning for managing non-compliance with clinical guidelines, in Proc. Workshop on CBR in the Health Sciences, International Conference on Case Based Reasoning (2007).
[38] M. d'Aquin, J. Lieber, A. Napoli, Adaptation knowledge acquisition: a case study for case-based decision support in oncology, *Computational Intelligence* **22** (2006), 161-176.
[39] C. Marling, D. Whitehouse, Case-based reasoning in the care of Alzheimer's disease patients, in Proc. International Conference on Case-based Reasoning (2001), 702-715.
[40] D.W. Bates et al., Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality, *JAMIA* **10** (2003), 523-530.

[41] M. Peleg, S. Tu, A. Manindroo, R.B. Altman, Modeling and analyzing biomedical processes using workflow/petri net models and tools, in Proc. Medinfo (2004), 74–78.

[42] L. Dazzi, C. Fassino, R. Saracco, S. Quaglini, & M. Stefanelli, A Patient workflow management system built on guidelines, in Proc. American Medical Informatics Annual Symposium (1997).

[43] D.Wang, M.Peleg, D.Bu, M.Cantor, G. Landesberg, E. Lunenfeld, S.Tu, G.E.Kaiser, G.Hripcsak, V.Patel, E.H.Shortliffe, Gesdor a generic execution model for sharing of computer-interpretable clinical practice guidelines, in Proc. American Medical Informatics Annual Symposium (2003), 694–698.

[44] S. Montani, P. Terenziani, A. Bottrighi, Supporting therapy selection in computerized clinical guidelines by means of decision theory, in Proc. Medinfo (2007), 855-860.

[45] S. Quaglini, M. Stefanelli, A. Cavallini, G. Micieli, C. Fassino, C. Mossa, Guideline-based careflow systems, *Artificial Inteligence in Medicine* **20(1)** (2000), 5-22.

[46] L. Ohno-Machado, J.H. Gennari, S. Murphy, N.L. Jain, S.W. Tu, D.E. Oliver, et al., The GuideLine Interchange Format: a model for representing guidelines, *JAMIA* **5(4)** (1998), 357-372.

[47] M. Peleg, A.A. Boxwala, et al., GLIF3: the evolution of a Guideline Representation Format, in Proc. American Medical Informatics Annual Symposium (2000).

[48] C. Gordon, S.I. Herbert, P. Johnson, Knowledge representation and clinical practice guidelines: the DILEMMA and PRESTIGE projects, in Proc. Medical Informatics Europe (1996), 511-515.

[49] E. Sherman, G. Hripcsak, J. Starren, R. Jender, P. Clayton, Using intermediate states to improve the ability of Arden Syntax to implement care plans and reuse knowledge, in Proc. Annual Symposium on Computer Applications in Medical Care (1995), 238-242.

[50] G. Duftschmid, S. Miksch, W. Gall, Verification of temporal scheduling constraints in clinical practice guidelines, *Artificial Intelligence in Medicine* **25(2)** (2002), 93-121.

[51] L. Anselma, P. Terenziani, S. Montani A. Bottrighi, Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines, *Artificial Intelligence in Medicine* **38(2)** (2006), 171-195.

[52] Y. Shahar, A framework for knowledge-based temporal abstraction in clinical domains, *Artificial Intelligence* **90(1)** (1997), 79-133.

[53] R. Kosara, S. Miksch, Metaphors of movement: a visualization and user interface for time-oriented, skeletal plans, *Artificial Intelligence in Medicine* **22(2)** (2001), 111-131.

[54] J. C. Augusto, Temporal reasoning for decision support in medicine. *Artificial Intelligence in Medicine* **33(1)** (2005), 1-24.

[55] P.L. Miller, *Expert critiquing systems: practice based medical consultation by computer*, Springer-Verlag, New York, 1986.

[56] J. van der Lei, Critiquing based on computer stored medical records, PhD thesis, Erasmus University Rotterdam, 1991.

[57] R. Sips, L. Braun, N. Roos, Applying intention-based guidelines for critiquing, in Proc. Workshop on AI techniques in healthcare: evidence-based guidelines and protocols, ECAI 2006

[58] Y. Shahar, S. Miksch, P. Johnson, A task-specific ontology for the application and critiquing of time oriented clinical guidelines, in Proc. Conference on Artificial Intelligence in Medicine in Europe (1997), 51–61.

[59] A. Advani, K. Lo, Y. Shahar, Intention-based critiquing of guideline-oriented medical care, in Proc. American Medical Informatics Annual Symposium (1998), 483–487.

[60] A. Advani, Y. Shahar, M.A. Musen., Medical quality assessment by scoring adherence to guideline intentions, *JAMIA* **9** (2002), S9– S97.

[61] M. Peleg, S. Tu, Comparing models of decision and action for guideline-based decision support: a case-study approach, Stanford University, 2002, Report No.: SMI-2002-0922.

[62] W. Aigner, S. Miksch, CareVis: Integrated visualization of computerized protocols and temporal patient data., in Proc. Workshop on Intelligent Data Analysis in Medicine and Pharmacology, 2004 .

[63] R. Kosara, S. Miksch, H. Hauser, Focus and context taken literally, *IEEE Computer Graphics and its Applications* **22(1)** (2002), 22-29.

[64] K. Bradbrook, G. Winstanley, D. Glasspool, J. Fox, R. Griffiths, AI Planning Technology as a Component of Computerised Clinical Practice Guidelines, AIME 2005, LNAI 3581 (2005), 171-180.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-121

Chapter 7
Adaptation of Clinical Practice Guidelines

Perry GROOT [1], Arjen HOMMERSOM, and Peter LUCAS
Radboud University Nijmegen
Toernooiveld 1, 6525 ED Nijmegen, the Netherlands

Abstract. A rigorous development process of clinical practice guidelines through a systematic appraisal of available evidence is costly and time consuming. One way to reduce the costs and time, and avoid unnecessary duplication of effort of guideline development is by relying on a local adaptation approach of guidelines developed at the (inter)national level by expert groups. In this chapter we survey the work on guideline adaptation, which includes methodologies, case studies, assessment of effectiveness, and related work on guideline adaptation in the Artificial Intelligence community.

Keywords. Protocol, Guideline development, Refinement

Introduction

The trend of the last decades has been to base clinical decision making more and more on sound scientific evidence, i.e., *evidence-based medicine* [78]. In practice this has led medical specialists to develop evidence-based clinical practice guidelines (CPGs) for promoting standards of medical care. Worldwide, a number of organisations, such as CBO[2] (Dutch Institute for Health Care Improvement) in the Netherlands and SIGN[3] (Scottish Intercollegiate Guidelines Network) in Scotland, have been founded to assist specialist groups and general practitioners in the development of guidelines. In 2002 the Guidelines International Network[4] was founded to promote systematic development of CPGs through international collaboration [51]. A rigorous development process of CPGs through a systematic appraisal of available evidence is, however, costly and time consuming.

One way to reduce the costs and time, and avoid unnecessary duplication of effort of guideline development is by relying on local adaptation of guidelines developed at the (inter)national level by expert groups. In this context ‘guideline adaptation’ is a process in which existing guidelines are modified to reflect the local situation so that they can be used within a different care setting. A local adaptation of one or more CPGs is often

[1] Corresponding Author: Perry Groot, Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, the Netherlands; E-mail: perry@cs.ru.nl.

[2] `http://www.cbo.nl` [accessed January 2008]

[3] `http://www.sign.ac.uk` [accessed January 2008]

[4] `http://www.g-i-n.net/` [accessed January 2008]

called *(clinical) protocol*. A protocol typically provides detailed information about duration, dose, or procedure, suited to the local context, that has been omitted from the original guideline. Basically, a medical protocol is a summary of the most important sections that are in a guideline, mostly recommendations, supplemented with hospital-specific details, although certain recommendations may be changed if they do not fit the local context. Several reasons may exist for adapting the recommendations of an guideline to suit a local context, e.g., cultural differences [66,65,55], constraints on resources [57], end-user involvement, etc. Legitimate changes can be made in recommendations even when the evidence they are based on is the same [17,35,7,77].

This book chapter is structured as follows. In Section 1 we discuss two methodologies that have appeared for the identification of candidate guidelines for local adaptation. In Section 2 we give an overview of case studies performed on guideline adaptation in terms of their objective for guideline adaptation, the setting, and the adaptation steps followed. In Section 3 we discuss a few randomised trials that focus on the effectiveness of the local adaptation approach on the uptake of nationally produced evidence-based CPGs. In Section 4 we discuss work done in the Artificial Intelligence community on guideline adaptation. We focus on 1) the adaptation of guidelines modelled in a formal representation language, 2) a logical representation of guidelines and theory refinement, and 3) machine learning techniques for learning and adapting guidelines from data. In Section 5 we give our overall conclusions on this chapter and Section 6 discusses some of the problems encountered in guideline adaptation that still need to be addressed.

1. Methodology

Guideline adaptation should follow similar procedures used in guideline development, including making transparent any decisions and key factors that influence the modifications. Two approaches have appeared for the identification of candidate guidelines for local adaptation, which are partly overlapping. The Practice Guideline Evaluation and Adaptation Cycle (PGEAC) [27,29,30,31] is a ten step approach (Figure 1), which can

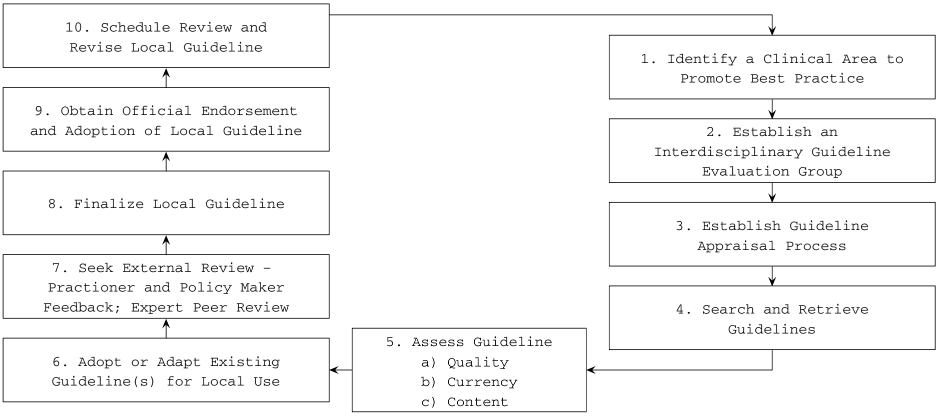

10. Schedule Review and Revise Local Guideline
9. Obtain Official Endorsement and Adoption of Local Guideline
8. Finalize Local Guideline
7. Seek External Review - Practioner and Policy Maker Feedback; Expert Peer Review
6. Adopt or Adapt Existing Guideline(s) for Local Use
5. Assess Guideline
a) Quality
b) Currency
c) Content
1. Identify a Clinical Area to Promote Best Practice
2. Establish an Interdisciplinary Guideline Evaluation Group
3. Establish Guideline Appraisal Process
4. Search and Retrieve Guidelines

Figure 1. The Practice Guidelines Evaluation and Adaptation Cycle (PGEAC) [27]. A methodology for the evaluation and adaptation of clinical practice guidelines.

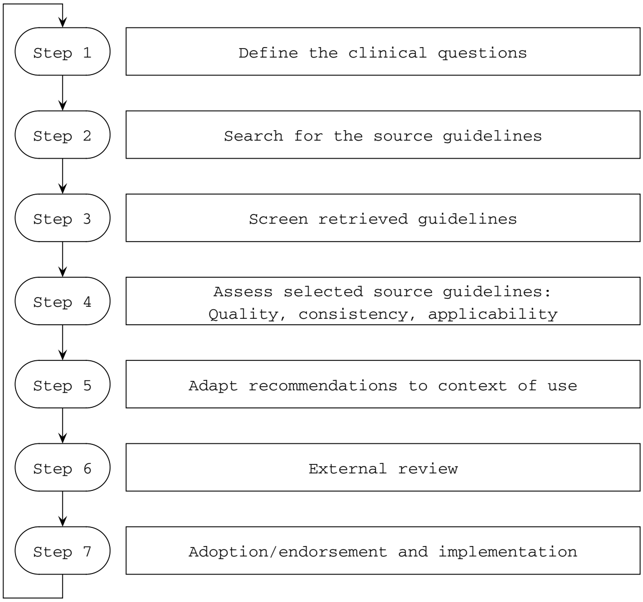

Step 1
Define the clinical questions
Step 2
Search for the source guidelines
Step 3
Screen retrieved guidelines
Step 4
Assess selected source guidelines:
Quality, consistency, applicability
Step 5
Adapt recommendations to context of use
Step 6
External review
Step 7
Adoption/endorsement and implementation

Figure 2. The procedure for guideline adaptation proposed by the ADAPTE working group [19].

be used to adopt a guideline with all its recommendations; adopt one guideline, but omit some recommendations that lack strong evidence or cannot be adopted locally; or take the best recommendations from several guidelines and adapt them to include them into one guideline.

The other approach (Figure 2) has been developed by the international working group ADAPTE[5] [19,20] and overlaps with the PGEAC approach. According to [20], the ADAPTE process was designed to create the conditions necessary to ensure the quality and validity of the resulting guideline and to foster adherence and ownership of professionals towards the adapted guideline whereas the PGEAC was designed to facilitate comparison of different guidelines and guideline recommendations on the same topic and offers a systematic way to evaluate guideline quality and clinical utility. Nevertheless, both approaches are fairly similar. We will discuss both approaches in more detail below.

1.1. Getting Started

The first two steps in the PGEAC approach is the identification of a clinical area to promote best practice and the establishment of an interdisciplinary guideline evaluation group. The identification of an area in which to promote best practice can be selected based on several reasons. These include the prevalence of the condition or its associated burdens, concerns about variations in care, associated costs of different care options, effectiveness of the guideline in influencing health care practice, the desire to keep care practice evidence-based, or the knowledge of the existence of evidence-based guidelines [29]. The establishment of the guideline evaluation group should comprise stakehold-

[5] `http://www.adapte.org` [accessed January 2008]

ers who will be affected by the guideline recommendations that will be selected. The multidisciplinarity of the group will enhance the relevance for practice and foster broad ownership and uptake of the adapted guideline [32] (cf. Section 3).

In the ADAPTE process, the need for developing a guideline and the establishment of an evaluation group are more considered as necessary conditions for starting the adaptation process and are not labelled explicitly as steps in the process. Additionally, however, the ADAPTE process focusses on defining a number of clinical questions, which is made explicit by using the PIPOH criteria: Patient population (including disease characteristics), Intervention(s) of interest, the Professionals to whom the guideline will be targeted, health Outcome(s) of interest, and the Health care setting in which the adapted guideline will be used.

1.2. Establish Guideline Appraisal Process

A guideline appraisal instrument needs to be chosen such that guidelines can systematically be assessed and compared according to the same criteria. Many appraisal instruments have been developed over the years [28], but the Appraisal of Guidelines Research and Evaluation (AGREE) instrument[6] is rapidly becoming the gold standard in guideline appraisal instruments [11]. The AGREE instrument was designed to assess the quality of the development process and the way it is reported. Hence, a rigorously developed guideline may still score insufficiently using the AGREE instrument when the development process is not described in detail.

1.3. Search for and Retrieve Guidelines

Both the PGEAC approach and the ADAPTE process give the following advice. To make sure that the most relevant high quality guidelines are obtained, a systematic search needs to be done which should start with guideline clearinghouses, e.g., the National Guideline Clearinghouse, the Guidelines International Network, or with country-specific databases. Additionally, websites of known guideline developers or search engines can be useful. For this, the population and intervention terms made explicit using PIPOH by the ADAPTE approach could be of help in the search strategy. The PIPOH approach is in fact very similar to the PICO approach, which involves Population, Intervention, Control or context, and Outcomes of interest [37], used in the PGEAC approach, but stated less explicitly.

In addition to the retrieval of guidelines, the ADAPTE process has an explicit step in which the retrieved guidelines are screened against the clinical questions defined earlier. Only those guidelines that correspond to the clinical questions are selected for a more detailed appraisal. Screening of guidelines is not part of the PGEAC approach, although it is suggested that additional criteria can be used in the search process to omit certain guidelines from the search results and only those guidelines that meet the minimum inclusion criteria will be used in the appraisal process.

[6] `http://www.agreecollaboration.org` [accessed January 2008]

1.4. Assess Guidelines

A pivotal step in the adaptation process is the appraisal of the guidelines. Both PGEAC and ADAPTE consider a number of fairly similar dimensions in the appraisal of the guidelines, i.e., the overall quality of the guideline, the consistency and currency of the guideline, and applicability of the guidelines recommendations to the context of use. The overall quality of the guideline can be used to identify the higher quality evidence-based guidelines, which can be used to restrict the number of guidelines that will follow a full appraisal when the appraisal of all retrieved guidelines is impractical. The consistency and currency of the guideline is validated by checking whether the guidelines recommendations are consistent with the cited evidence and whether these recommendations are still current or need to be updated according to newly obtained results. Finally, each guideline needs to be compared in terms of the recommendations made and level of evidence supporting the recommendations and whether they are applicable to the context of use.

1.5. Adopt or Adapt Guidelines for Local Use

After the appraisal, one can adopt or adapt existing guidelines. Adopting a guideline means choosing the best guideline and accepting all its recommendations. Adapting guidelines means taking the best recommendations from several guidelines, applicable to the local context, and adapting and reformatting them into a new guideline. Strong evidence-based recommendations should only be changed when the supporting evidence has changed or when not applicable to the local context, for example, because of resource constraints [57]. The ADAPTE process points out that it is still possible to consider de novo development of a guideline.

1.6. External Review

Before the dissemination and implementation of the resulting draft of local recommendations, it should be sent to local practitioners, organisational policy makers, and other stakeholders for a review. This also holds for de novo guideline development and the recommendations in the PGEAC and ADAPTE approach are, therefore, identical.

1.7. Adoption and Implementation

In this phase the same issues hold for guideline adaptation as for guideline development. PGEAC and ADAPTE therefore give similar recommendations. When the guideline has been finalised, official endorsement from policy makers should be sought for those clinical care settings in which the guideline will be implemented. The formal decision making and procedural process for endorsing the guideline should be documented by the organisation and a dissemination and implementation plan should be finalised.

1.8. Scheduling Review and Revision of Local Guideline

In contrast with ADAPTE, PGEAC explicitly mentions that the adaptation of guidelines is a process cycle and that revisions of the local guideline need to be scheduled. This aspect has had less attention than guideline development, but several criteria (e.g., expiry date, changes in evidence, important outcomes, availability of health care resources, new interventions, etc.) can be set to determine when and what should be reviewed and updated [9,63].

1.9. Final Remarks

Summarising, both the PGEAC cycle and ADAPTE process are fairly similar methodologies for guideline adaptation. Some differences exist, but are mainly differences in explicitness.

Both groups have recently merged into the ADAPTE group[7] whose main endeavour is the development and validation of a generic guideline adaptation process that fosters the validity and quality as well as the users' sense of ownership toward the adapted guideline. This has been coined the ADAPTE framework and has also resulted in a generic manual and resource toolkit for guideline adaptation. Both are, at the time of writing, still undergoing an evaluation study.

2. Case Studies

A number of publications have appeared that report practical examples and experiences with guideline adaptation. In this section we give an assessment of these publications in terms of their objectives for adaptation, the country in which the adaptation took place, and the steps followed in the adaptation process (cf. Table 1), based on a previous literature survey reported in [19].

2.1. Alternative to de Novo Guideline Development

Several publications have appeared that consider guideline adaptation as an alternative to de novo guideline development [43,30,31,76]. The goals in these publications are loosely formulated as the need for providing evidence-based care in a certain medical area, but without weighing all the pros and cons of guideline development versus guideline adaptation. The pros for guideline adaptation are more or less taken for granted. Additionally, those reports focus on the applicability of a guideline adaptation process. For example, [30,31] use the PGEAC approach (cf. Section 1) whereas [43] uses a guideline adaptation process established by the Registered Nurses Association of Ontario, Canada, in 1999. The guideline adaptation process is therefore overall well documented and covers almost all the guideline adaptation steps included in the review criteria. Each of the reported adaptation processes started with a search for relevant CPGs whereas other reports started their adaptation process from a guideline already selected. These reports are also the only ones to include a detailed assessment of the quality of the contents of the relevant guidelines. In [43,30,31] the quality of the guidelines is established using the Appraisal Instrument for Clinical Practice Guidelines [10], which is an older version of the AGREE instrument [11] used by [76] for quality assessment.

[7] `http://www.adapte.org` [accessed January 2008]

Table 1. Case-study descriptions on guideline adaptation from [19].

Reference	Country	Adaptation Process Steps					
		A	B	C	D	E	F
Adaptation as an alternative to de novo development							
Graham *et al.*, 2002 [30]	Canada	+	+	+	+	+	+
Graham *et al.*, 2005 [31]	Canada	+	+	+	-	+	-
Macleod *et al.*, 2002 [43]	Canada	+	+	+	-	+	+
Voellinger *et al.*, 2003 [76]	Switzerland	+	+	+	+	+	-
Adaptation as part of an implementation process (international):							
Armstrong *et al.*, 2004 [2]	Canada	-	-	+	-	+/-	-
Croudace *et al.*, 2003 [12]	UK	-	-	-	-	-	+
De Wit *et al.*, 2000 [16]	Europe	-	-	+	-	+	+
Glasier *et al.*, 2003 [26]	UK	-	-	+	-	-	-
Hungin *et al.*, 2001 [39]	Europe	-	-	+	-	-	-
Peleg *et al.*, 2006 [54]	US/Israel	-	-	+	-	-	+
Reddy *et al.*, 1999 [56]	India	-	-	+	-	-	+
Rhineart *et al.*, 1991 [57]	Indonesia	-	-	+	-	-	-
Shye *et al.*, 2000 [67]	US/Israel	-	-	+	-	+	-
Adaptation as part of an implementation process (national):							
Brown *et al.*, 1995 [6]	US	-	-	+	-	+	-
Capdenat *et al.*, 1998 [8]	France	-	-	+	-	-	+
Hall *et al.*, 2000 [34]	UK	-	-	-	-	+	+
Lobach, 1995 [42]	US	-	-	+/-	-	+	-
Maviglia *et al.*, 2003 [47]	US	-	-	-	-	+	+
Silagy *et al.*, 2002 [68]	Australia	-	-	+	-	+	+
Tomlinson *et al.*, 2000 [73]	UK	+	-	-	+	+	+

Adaptation process steps: A) Search for and retrieve existing guidelines, B) Assess guidelines, C) Adopt or adept for local use, D) Complementary literature search, E) Seek external review, and F) Implementation. These steps correspond accordingly to steps in the PGEAC approach: A-4, B-5, C-6, E-7, and F-9.

2.2. *Adaptation as Part of an Implementation Process*

Other publications that reported experiences on the process of guideline adaptation, were usually given in the context of an implementation process of a guideline at a local site. This was done either by adapting an international guideline developed in a different country [2,12,16,26,39,54,56,57,67] or by adapting a national guideline to a local context [6,8,34,42,47,68,73].

For example, [57] adapts a CDC guideline for the prevention of nosocomial infection in a pediatric intensive care unit in Jakarta, Indonesia. Because of limited resources, changes had to be made to the CDC guideline as well as the local environment. For example, the installment of handwashing sinks, avoiding use of critical devices, indirect quality control of sterilisation by monitoring time and temperature, etc. Whenever possible, a low-technology, common sense approach was used such that fundamental infection control principles could be preserved without straining the local resources and capabilities.

All reviewed publications that focus on implementation basically follow the same adaptation steps, which include the adaptation of the guideline in question, the implementation, and, in about half the cases, an external review. The other process steps were almost never included in the report. The guideline to be adapted was already assumed to be given and no search for guidelines was performed. Also, the assessment of the guideline was lacking in all the reviewed publications although this is considered to be a pivotal step in the adaptation process by both the PGEAC and ADAPTE guideline adaptation approach (cf. Section 1). This seems to indicate that work on guideline adaptation is still very much in development and case studies are more of an empiric nature. This is also supported by the fact that out of the twenty case studies investigated, only five were performed before the year 2000.

2.3. Guideline Integration with Local Decision Support

Besides the publications presented in Table 1, many other publications exist that focus on the integration of medical guidelines with a local decision support system, but that do not emphasise guideline adaptation. Nevertheless, in many of such cases technical issues are addressed when implementing guidelines at a local site, which may result in changes compared to the original guideline (e.g., the clinical information system in use, the data models of the electronic medical record (EMR), and the data actually collected).

Several groups have reported their experience on guideline adaptation when implementing them at a local site. For example, Shiffman [64] investigated the validity of an asthma guideline through a logical analysis showing that the guideline under study was incomplete and ambiguous. Such logical integrity violations need to be addressed before the guideline can be operationalised and a structured data entry system can be devised through an examination of the guideline decision points. [72] investigates the practical considerations when adapting an inpatient heart failure guideline to the outpatient setting and implementing it within an EMR. As a result, about one-third of the original guideline recommendations were not included in the final implementation, because of a different setting at the local site. Additionally, some guideline data definitions had to be translated into several EMR entries as the data definitions were not directly available from the local EMR. Similar results are reported in [54], for the adaptation and implementation of an American diabetes foot care guideline at an Israelian site.

Some principled approaches are being advocated that may be used to overcome difficulties when integrating guidelines with local decision support systems. For example, [60,25] advocate resorting to guidelines intentions, in order to ensure the adaptability of the procedure to different contexts, while still preserving the original intentional objectives. A setting-independent format is advocated by [4], which relies on an explicit description of dependencies between actions, and requires that they will be preserved by adaptation. Argumentation is advocated by [24] to display arguments in favour or against a certain treatment from multiple sources in order for the physician to make an educated decision.

These studies show that when adapting a guideline for a local site, one should consider the implementation and integration with the local setting (e.g., a local EMR) as soon as possible, as this may have major impacts for the encoding (i.e., the computational representation). Such issues are currently, however, not yet supported by guideline adaptation methodologies (cf. Section 1).

3. Assessment of Effectiveness of Local Adaptation on Uptake of Guidelines

Guideline development at a national level by multidisciplinary groups of experts with adaptations being made at a local level is already part of the process of several organisations (e.g., CBO, SIGN). Advocates of this approach argue that evidence-based guidelines can be developed at the national level by expert groups as the skills necessary are available at this level, but unlikely to be available at a local level [68]. Although several arguments can be given in favour of an approach of local adaptation of guidelines developed at a national level by clinical experts, so far this approach has had very little formal evaluation [68].

A few randomised trials have appeared in the literature that specifically focus on the effectiveness of the local adaptation approach on the uptake of nationally produced evidence-based CPGs [68,12]. For example, in [12] 30 (out of 42) practices from Bristol, UK were screened for some time before they were split into two groups of 15 practices, containing 56 and 60 GPs each. One group continued with the usual care while the other group adapted the WHO ICD-10 PHC guidelines. Both practices were then screened again using 186 patients in each group.

Both randomised trials [68,12] report no significant changes in practitioner behaviour or patient outcomes, which seems to contradict earlier reports that involvement of end-users in the development process may lead to an increased uptake of CPGs [32]. It is a well known problem, however, that CPGs - without any adaptation - often fail to affect clinical practice [33,79]. Systematic research is therefore done to find out why physicians do not follow CPGs [14,13,36]. (The issue of physicians' compliance is also discussed in detail in Chapter 9.) These complications, as well as limitations in the performed trials (e.g., a minimal rigorous assessment of the evidence and appraisal of additional literature), limit the results of the studies in resolving the effectiveness of local adaptation on the uptake of clinical guidelines.

4. Adaptation and Artificial Intelligence

So far, we looked at guideline adaptation mostly from the view of the medical community. In this chapter, we relate guideline adaptation to concepts in Artificial Intelligence (AI). As not much work has yet been done in AI that specifically focusses on guideline adaptation, many of the things we discuss here will be from a possible future research perspective.

Guideline development and guideline adaptation is a knowledge engineering task that can be subdivided into various phases such as knowledge acquisition, representation design, implementation, evaluation, and re-implementation [70]. Most of the modelling activities can be carried out by tools as well as an engineer, i.e., the concept of *balanced cooperation* [49]. Several tools have already been built for guideline formalisation based on knowledge acquisition (KA) and information extraction (IE) techniques such as Stepper, GEM-Cutter, DELT/A, Uruz, AsbruView, Protégé, AREZZO, and TALLIS [40] (cf. Chapter 8). Such tools usually take the text document of a medical guideline as starting point, from which a formal model is derived.

IE is an emerging technology in natural language processing to locate facts and specific pieces of information from unstructured natural text, which can either be developed using a knowledge engineering approach or an automatic learning approach. The automatic approach takes as input a set of documents in natural language and outputs a set of extraction patterns using machine learning techniques [40]. Below we discuss these topics in more detail. Firstly, we look at formal guideline representation languages developed in the Artificial Intelligence (AI) community. Secondly, we look at adaptation from a logic viewpoint as a theory refinement problem. Thirdly, we look at machine learning techniques for developing and adapting guidelines.

4.1. Guideline Representation Language

Researchers in AI have been working toward offering computer-based support in the development and deployment of guidelines by using computer-oriented languages and tools [15,53]. Examples of languages include PRO*forma* [22,23], Asbru [59,61], EON [74,75], and GLIF [52] (cf. Chapter 2). These languages model complex clinical processes as a 'network of tasks', where a task consists of a number of steps, each step having a specific function or goal [21,52]. Adaptation of medical guidelines is therefore often considered a form of program refinement or program transformation in the AI community. We discuss several studies on adaptation of a guideline represented in a formal guideline representation language in more detail below. For details concerning the languages, the reader is referred to Chapter 2 of this book.

As formal guideline representation languages have been evolving since the 1990s, only a few case studies [46,38,45,54] were found that report on the adaptation of a formal model written in a formal guideline representation language in the context of an adaptation process. (References [38,54] are part of this book.)

The studies reported in [46,38,45] use the Asbru language, whereas the study reported in [54] uses the GLIF language. The work of [38] does not focus on the adaptation process itself, but focusses on the verification of the begin- and end-product of the adaptation process for obtaining differences between guideline and protocol using a formal approach. The differences between the same guideline and protocol are also analysed in [46] from an informal angle for the guideline and protocol text and for the corresponding Asbru models.

According to [46], the most frequent occurring differences between the guideline and protocol text are refinements of the guideline recommendations in which elements are made more specific or substituted to provide more detail about treatments. For example, the protocol may specify the therapy of choice in cases where the guideline offers different alternatives, or the protocol may include special cases not considered in the guideline. Other refinements, analysed in [46] were found to be the result of recent evidence, i.e., the protocol was more up-to-date than the guideline from which it was adapted. Overall, most of the guideline and protocol text were found to be similar, although a few sections seemed different, because of a different layout used for the protocol.

An analysis of differences between the constructed Asbru models of the guideline and protocol showed similar results. Also on this level it was found that much of the Asbru model of the guideline could be reused to construct the Asbru model of the protocol. These results are in agreement with the results of [54], which concludes that a significant portion of the original guideline was also useful for the local site, although the local adaptation process also had significant effects on parts of the encoding.

4.2. Logical Modelling of Guideline Adaptation

The task network modelling languages of the previous section are not suitable to define adaptation in such a way that one can *reason about the adaptation process itself.* Although a lot of work has already been done about formal verification of CPGs (cf. Chapter 4), such work was never done primarily in the task network modelling language, but always in some meta-language. Furthermore, the properties typically looked at (e.g., termination, reachability of plans, etc.) say nothing about the adaptation *process*, but merely something of the final product of guideline adaptation (cf. [38]).

In this section, we look at first order logic as a meta-language for describing the guideline adaptation process. Let T be some theory that represents the formalisation of the guideline text whereas T' represents some adaptation from T. More generally, T' can be identified with a theory T_i in an adaptation process

$$T \equiv T_0 \Rightarrow T_1 \Rightarrow \cdots \Rightarrow T_n$$

in which each theory T_i is some adaptation step of the original guideline T (with '$\Rightarrow$' not necessarily being material implication). Furthermore, T should have an adaptable representation, i.e., the theory remains consistent when local information (e.g., a particular choice between certain resources) is added to the theory:

$$T \cup LI \not\models \bot$$

for any piece of local information LI. Local adaptability will not hold in general for guideline adaptations as some piece of local information has already been used to adapt the guideline, i.e.,

$$\exists LI \quad T_i \cup LI \models \bot$$

These are just some thoughts on the characterisation of the guideline adaptation process and is far from being complete.

In guideline adaptation, we may consider the guideline to be a theory that provides solutions (i.e., treatment paths) for some domain and the protocol to be a revision of this theory. Many reasons may exist for revising the guideline, e.g., local restrictions are invalidated, newly obtained evidence provides new patient management options, financial costs of drug or equipment manufacturing has decreased, or additional formatting is needed to increase readability. In this light, guideline adaptation may be considered a *theory refinement* problem and current research on refining knowledge-based systems may offer interesting possibilities for guideline development and guideline adaptation.

Theory Refinement

In general, whenever a theory is built of some real-world application domain, one sooner or later is confronted with the problem of maintenance of the model. As building up a theory of a domain is very time consuming, rebuilding the theory from scratch, each time the application changes, is usually too costly. Hence, one would like to detect shortcomings of the theory and make repairs to the model. Several reasons may be distinguished for wanting to change the theory [81,70]:

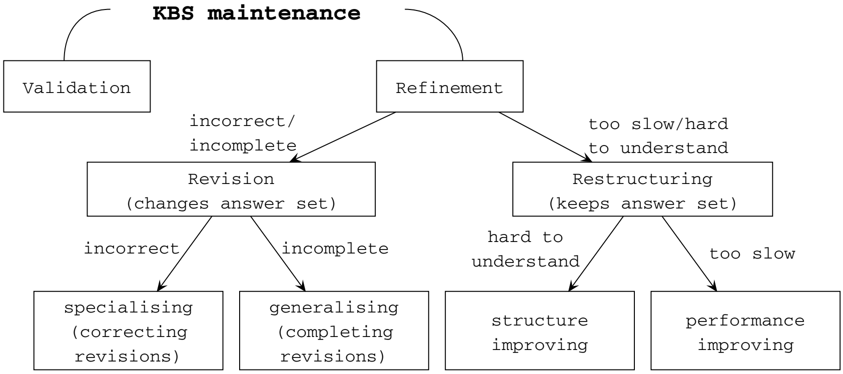


Figure 3. KBS maintenance (adapted from [81,70]).

Revision: The theory gives wrong answers, either because the theory does not cover all cases, or some cases are covered incorrectly.

Performance enhancement: The theory provides solutions to cases that are too long or too costly as they can be improved.

Restructuring: The theory has become ill structured and is not transparent, because of, for example, redundancies and implicit concepts. Although the theory provides correct answers, they are not open for human inspection and explanations are incomprehensible.

This is also graphically represented as part of Figure 3. The maintenance task consists of three closely related topics. *Validation* of a KBS is concerned with determining whether the formal model of reality, i.e., the knowledge base, does indeed correspond with reality. *Revision* of a KBS is concerned with modifying its answer set to deal with the inconsistency or incompleteness of the knowledge base. *Restructuring* of the KBS is concerned with changing the representation of the knowledge base to, for example, increase speed or readability, without changing the answer set.

4.3. Machine Learning

Here, we take a closer look at machine learning techniques. In particular, machine learning techniques for learning logical theories and logical relations among concepts such as relational learning and inductive logic programming techniques. Furthermore, we do not restrict the input to a set of documents in natural language. We focus on the integration of machine learning methods into the modelling environment of the knowledge engineer to induce rules from examples, possibly hand-tailored to experts' specifications. Besides IE, machine learning techniques are also applicable to guideline development and adaptation by building up the concepts underlying medical guidelines from medical data. Since the 1990s, machine learning techniques have been gaining importance in a knowledge engineering context [50,3,71] and have resulted in a number of tools such as MOBAL [50] and LINK [69]. Below we discuss in more detail the research and the insights gained in this research area and relate it to guideline adaptation.

Inductive Learning Algorithms

Here, we give a short overview of existing inductive learning algorithms and the issues involved in developing such algorithms. The input language to the learning algorithm is an important aspect when comparing different approaches. Although many approaches use some specific input language, some general classes have been identified, i.e., propositional, structured objects (i.e., binary relations of depth one; e.g., ARCH, Induce, Cluster, KHG), and restricted first-order theories (e.g., KL-ONE, Horn Logic, FOIL, GOLEM, RDT, Inductive Logic Programming (ILP) techniques) [41,1]. Any of these domains can in principle be used for learning, however, intuitive relational connectedness between concepts is lost, for example, when using a propositional input language.

Earlier learning algorithms often produce outputs that are incompatible with their inputs, however, much research has focused on finding a common ground for in- and output. Motivation for this is to have a 'closed-loop' learning process [18], i.e., allowing the knowledge engineering task to be done incrementally by incorporating the output into subsequent input. Examples of such efforts are attribute arrays, Prolog facts or clauses, or representation languages for interchanging information between several algorithms such as CKRL and KIF [70]. More often than not, the output language can be further restricted to a true subset of the input language. Making this explicit has been considered an important component in ILP research.

Many relational learning algorithms, however, assume that their background knowledge is static, i.e., the knowledge does not change during learning [80]. This, of course, limits their applicability w.r.t. guideline adaptation or updating them on a regular basis. A few systems have been developed that do support incremental learning with changing background knowledge such as AUDREY, which can explicitly revise its domain theories, and FOCL, which can recover from incomplete and incorrect domain theories [1]. An algorithm that solves the problem of incrementally building up expertise in a rapidly changing environment is an interesting research area in developing relational learning algorithms and might greatly benefit guideline development and adaptation.

Machine learning techniques and relational learning algorithms offer interesting possibilities for guideline development and guideline adaptation. Much work, however, still needs to be done in this research area to make learning algorithms effective tools for such applications. For example, supporting incremental learning with changing background knowledge is still largely an open problem. Nevertheless, some approaches are starting to appear that diverge from the current mainstream view of guideline development and formalisation of text documents using formal representations and data mining techniques (e.g., [44,82,58]).

4.4. Conclusions

So far, research on guideline adaptation has mainly focussed on the adaptation of documents without considering any computational representation. Early studies show that computational representations are also likely to be adaptable and that large portions are also useful for the local site. A shortcoming of this work is, however, that it is difficult (or impossible) to give general statements about adaptation of guidelines in terms of task network modelling languages and one is therefore unable to reason about the adaptation process. Other representations such as logic, allows one to state guideline development in more general terms and guideline adaptation as a theory refinement problem.

Machine learning techniques offer interesting research prospects, which diverges from the mainstream document oriented view of guidelines, for using logical or probabilistic representations for learning and adapting guidelines directly from clinical data.

5. Overall Conclusions

In this chapter we have given an overview of work done by the medical guideline community on guideline adaptation. We discussed two state of the art guideline adaptation methodologies, the PGEAC approach and ADAPTE process, which were found to be fairly similar, and both are now being merged into the ADAPTE framework. We also gave an overview of case studies done on guideline adaptation. Most of these studies were of an empirical nature and adapted guidelines without any adaptation methodology. We also noted that guideline adaptation by the medical community as part of an implementation process does not take into account any computable representation although work done in the AI community has shown that this may influence the adaptation of a guideline. We finished the part on guideline adaptation by the medical community by two assessment studies of the effectiveness of the local adaptation approach on the uptake of nationally produced evidence-based CPGs. Both studies, although inconclusive, were unable to show a positive effect in uptake through guideline adaptation.

Thereafter, we looked at work done by the AI community on guideline development and guideline adaptation. Recent work is very much focused on offering computer-based support in the development and deployment of guidelines by using computer-oriented languages and tools. Almost all current work takes a guideline text document as starting point for building a formal model in some task network modelling language using knowledge acquisition and information extraction techniques. Such techniques are limited for describing the adaptation process and some meta-language is therefore necessary. For this, we looked at first order logic and looked at guideline adaptation as a theory refinement problem. We went beyond the current approaches by considering the integration of machine learning techniques into the guideline development process for learning and adapting guidelines from data. There is a still a big gap between work done in the guideline community and work done in the AI community. Much work still needs to be done to bridge this gap.

6. Research Agenda

With respect to guideline adaptation, some of the main problems in the guideline community is inaccessibility of CPGs and their varying quality. Not all guidelines are published or accessible through the Internet [76]. Furthermore, several studies suggest that the quality of developed guidelines is highly variable and that often many details, which are needed for assessment, are missing [7,5,35,48,62]. For example, Hart *et al.,* [35] report that several stroke prevention guidelines provide no adequate methodologic information (panel selection, patient preferences, justification of risk stratification criteria) to permit assessment of their quality, potential bias, and clinical applicability. The management recommendations were found to be relatively consistent between guidelines, but differed in several important areas. Voellinger *et al.,* [76] report problems with guideline adaptation, because of a lack of guidelines focussing on co-morbidities as well as the need to adapt the various or missing levels of evidence to a uniform scale. The experience of the organisation involved in guideline adaptation is one factor that may explain the varying quality of CPGs. Training and instruments such as the AGREE instrument should be further developed to improve the quality of CPGs.

In the field of AI there is still a lot of work to be done as the topic of guideline adaptation is not well understood at this moment. There should be a clear characterisation of the guideline adaptation process - what kind of adaptation operations are allowed - before tools and languages can be build to support the adaptation task. Also, some people have the misguided belief that guidelines can be represented as executable plans. One might want to represent a protocol that can be executed by a local decision support system, but current evidence-based guidelines are often too incomplete to be considered an executable plan. Instead, CPGs are currently merely some constraints on executing clinical practice. How to combine general constraints with workflow management is still an open problem. Other languages than task network modelling languages may be needed. Finally, it is the belief of the authors that the adaptation process can only be rightly supported by integrating techniques earlier in the development process of CPGs. Current work, that takes the paper document as starting point, is flawed as the textual documents are currently missing too much detail to support all kinds of other tasks such as verification and adaptation. The bottleneck in developing guidelines is still the development process itself. Techniques such as machine learning offer interesting new possibilities that could help in alleviating the problems mentioned above.

References

[1] D.W. Aha. Relating relational learning algorithms. In S. Muggleton, editor, *Inductive Logic Programming*, pages 233–260. Academic Press, 1992.

[2] P.W. Armstrong, P. Bogaty, C.E. Buller, P. Dorian, and B.J. ONeill. The 2004 ACC/AHA Guidelines: a perspective and adaptation for Canada by the Canadian Cardiovascular Society Working Group. *The Canadian Journal of Cardiology*, 20:1075–1079, 2004.

[3] F. Bergadano, A. Giordiana, and L. Siatta. *Machine Learning: An Integrated Approach and its Application*. Ellis Horwood, Chichester, UK, 1991.

[4] A.A. Boxwala, Q. Zeng, D. Tate, R.A. Greenes, and D.G. Fairchild. Applying axiomatic design methodology to create guidelines that are locally adaptable. In *Proceedings American Medical Informatics Annual Symposium*, page 980, 2002.

[5] F. Bro and F.B. Waldorff. Guidelines – lets take a break and then move forward together! *Scandinavian Journal of Primary Health Care*, 22:2–5, 2004.

[6] J.B. Brown, D. Shye, and B. McFarland. The paradox of guideline implementation: how AHCPR's depression guideline was adapted at kaiser permanente northwest region. *Joint Commission Journal on Quality Improvement*, 21:5–21, 1995.

[7] J.S. Burgers, B. Fervers, M. Haugh, M. Brouwers, G. Browman, T. Philip, and F.A. Cluzeau. International assessment of the quality of clinical practice guidelines in oncology using the appraisal of guidelines and research and evaluation instrument. *Journal of Clinical Oncology*, 22(10):2000–2007, 2004.

[8] E. Capdenat Saint-Martin, P. Michel, J.M. Raymond, H. Iskandar, C. Chevalier, M.N. Petitpierre, L. Daubech, M. Amouretti, and P. Maurette. Description of local adaptation of national guidelines and of active feedback for rationalising preoperative screening in patients at low risk from anesthetics in a French University hospital. *QHC*, 7:5–11, 1998.

[9] E. Clark, E.F. Donovan, and P. Schoettker. From outdated to updated, keeping clinical guidelines valid. *International Journal for Quality in Health Care*, 18:165–166, 2006.

[10] F. Cluzeau, P. Littlejohns, J. Grimshaw, and G. Feder. *Appraisal instrument for clinical guidelines*. London: St. George's Hospital Medical School, 1997.

[11] The AGREE Collaboration. Development and validation of an international appraisal instrument for assessing the quality of clinical practice guidelines: the AGREE project. *Quality and Safety in Health Care*, 12(1):18–23, 2003.

[12] T. Croudace, J. Evans, G. Harrison, D.J. Sharp, E. Wilkinson, G. McCann, M. Spence, C. Crilly, and L. Brindle. Impact of the ICD-10 Primary Health Care (PHC) diagnostic and management guidelines for mental disorders on detection and outcome in primary care. cluster randomised controlled trial. *BJP*, 182:20–30, 2003.

[13] K. Dahlberg. Reasons physicians do not follow clinical practice guidelines. *The Journal of the American Medical Association*, 283(13):1686, 2000.

[14] D.A. Davis and A. Taylor-Vaisey. Translating guidelines into practice. a systematic review of theoretic concepts, practical experience and research evidence in the adoption of clinical practice guidelines. *Canadian Medical Association journal*, 157(4):408–416, 1997.

[15] P.A. de Clercq, J.A. Blom, H.H.M. Korsten, and A. Hasman. Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial Intelligence in Medicine*, 31(1):1–27, 2004.

[16] N.J. de Wit, J. Mendive, B. Seifert, F. Cardin, and G. Rubin. Guidelines on the management of h.pylori in primary care: development of an implementation strategy. *Family Practice*, 17(suppl2.):27–32, 2000.

[17] F. Eisinger, G. Geller, W. Burke, and N.A. Holtzman. Cultural basis for differences between us and french clinical recommendations for women at increased risk of breast and ovarian cancer. *Lancet*, 353:919–920, 1999.

[18] W. Emde, C. Habel, and C.R. Rollinger. The discovery of the equator or concept driven learning. In *IJCAI*, pages 355–458, 1983.

[19] B. Fervers, J.S. Burgers, M.C. Haugh, J. Latreille, N. Mlika-Cabanne, L. Paquet, M. Coulombe, M. Poirier, and B. Burnand. Adaptation of clinical guidelines: literature review and proposition for a framework and procedure. *International Journal for Quality in Health Care*, 18(3):167–176, 2006.

[20] B. Fervers and I.D. Graham. Guideline adaptation - the way forward? In *Guidelines International Network Conference 2005*, Lyon, 2005.

[21] J. Fox and S. Das. *Safe and Sound: Artificial Intelligence in Hazardous Applications*. AAAI Press, 2000.

[22] J. Fox, N. Johns, A. Rahmanzadeh, and R. Thomson. PRO*forma*: A method and language for specifying clinical guidelines and protocols. In J. Brender, J.P. Christensen, Scherrer. J.R., and P. McNair, editors, *Medical Informatics Europe*, pages 516–520. IOS Press, 1996.

[23] J. Fox, N. Johns, A. Rahmanzadeh, and R. Thomson. PRO*forma*: a general technology for clinical decision support systems. *Computer Methods and Programs in Biomedicine*, 54:59–67, 1997.

[24] J. Fox, V. Patkar, and R Thomson. Decision support for healthcare: the PRO*forma* evidence base. *Informatics in Primary Care*, 14:49–54, 2006.

[25] D.B. Fridsma, J.H. Gennari, and M.A. Musen. Making generic guidelines site-specific. In *Proceedings American Medical Informatics Annual Symposium*, pages 597–601, 1996.

[26] A. Glasier, S. Brechin, R. Raine, and G. Penney. A consensus process to adapt the world health organization selected practice recommendations for UK use. *Contraception*, 68:327–333, 2003.

[27] I. Graham, M. Harrison, and M. Brouwers. *Evaluating and adapting practice guidelines for local use: a conceptual framework*, pages 213–229. Churchill Livingstone, 2003.

[28] I.D. Graham, L.A. Calder, P.C. Hébert, A.O. Carter, and J.M. Tetroe. A comparison of clinical practice guideline appraisal instruments. *International Journal of Technology Assessment in Health Care*, 16:1024–1038, 2000.

[29] I.D. Graham and M.B. Harrison. Evaluation and adaptation of clinical practice guidelines. *Evidence-based nursing*, 8:68–72, 2005.

[30] I.D. Graham, M.B. Harrison, M. Brouwers, B.L. Davies, and S. Dunn. Facilitating the use of evidence in practice: evaluating and adapting clinical practice guidelines for local use by health care organizations. *Journal of obstetric, gynecologic, and neonatal nursing*, 31:599–611, 2002.

[31] I.D. Graham, M.B. Harrison, K. Lorimer, T. Piercianowski, E. Friedberg, M. Buchanan, and C. Harris. Adapting national and international leg ulcer practice guidelines for local use: the ontario leg ulcer community care protocol. *Advances in skin & wound care*, 18:307–318, 2005.

[32] J.M. Grimshaw and A. Hutchinson. Clinical practice guidelines – do they enhance value for money in health care. *British medical bulletin*, 51:927–940, 1995.

[33] J.M. Grimshaw and I.T. Russell. Effect of clinical guidelines on medical practice: a systematic review

of rigorous evaluations. *Lancet*, 342(8883):1317–1322, 1993.

[34] L. Hall and M. Eccles. Case study of an inter-professional and interorganisational programme to adapt, implement and evaluate clinical guidelines in secondary care. *Clinical performance and quality health care*, 8:72–82, 2000.

[35] R.G. Hart and R.D. Bailey. An assessment of guidelines for prevention of ischemic stroke. *Neurology*, 59:977–982, 2002.

[36] A. Haynes, B. Haines. Barriers and bridges to evidence based clinical practice. *BMJ*, 317(7153):273–276, 1998.

[37] C. Heneghan and D. Badenoch. *Evidence-based medicine toolkit*. Blackwell Publishing, Oxford, United Kingdom, 2006.

[38] A. Hommersom, P. Groot, P. Lucas, M. Marcos, and B. Martinez-Salvador. *Computer-based medical guidelines and protocols: A primer and current trends*, chapter A Constraint-based Approach to Medical Guidelines and Protocols. IOS Press, 2008.

[39] P. Hungin. Working together across international boundaries: improving the primary-secondary care network. *European journal of gastroenterology & hepatology*, 13(suppl2.):13–16, 2001.

[40] K. Kaiser, C. Akkaya, and S. Miksch. How can information extraction ease formalizing treatment processes in clinical practice guidelines? a method and its evaluation. *Artifical Intelligence in Medicine*, 39(2):151–163, 2007.

[41] N. Lavrač and S. Džeroski. Review of "inductive logic programming: Techniques and applications". *Machine Learning*, 23(1):103–108, 1996.

[42] D.F. Lobach. A model for adapting clinical guidelines for electronic implementation in primary care. In *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, pages 581–585, 1995.

[43] F.E. MacLeod, M.B. Harrison, and I.D. Graham. The process of developing best practice guidelines for nurses in Ontario: risk assessment and prevention of pressure ulcers. *Ostomy/wound management*, 48:30–38, 2002.

[44] S. Mani, C. Aliferis, S. Krishnaswami, and T. Kotchen. Learning causal and predictive clinical practice guidelines from data. In *Proceedings of MedInfo*. IOS Press, 2007.

[45] M. Marcos, B. Martínez-Salvador, M. Balser, J. Schmitt, A. Dunets, S. Bäumler, and W. Reif. A formal approach to the refinement of medical practice guidelines into protocols. *Submitted to Computers in Biology and Medicine*, 2006.

[46] M. Marcos, B. Martínez-Salvador, A. Hommersom, P. Groot, P. Lucas, A. Jovell, and S. Blancafort. Case-study in transformations for protocol development from guidelines. Technical Report D5.1, Protocure II, 2006.

[47] R.D. Mavliglia, S.M. adn Zielstorff, M. Paterno, J.M. Teich, D.W. Bates, and Kuperman G.J. Automating complex guidelines for chronic disease: lessons learned. *Journal of the American Medical Informatics Association*, 10:154–165, 2003.

[48] J. Miller and J. Petrie. Development of practice guidelines. *Lancet*, 355:82–83, 2000.

[49] K. Morik. Balanced cooperative modeling. *Machine Learning*, 11(2-3):217–235, 1993.

[50] K. Morik, S. Wrobel, J.U. Kiets, and W. Emde. *Knowledge Acquisition and Machine Learning*. Academic Press, London, 1993.

[51] G. Ollenschläger, C. Marshall, S. Qureshi, K. Rossenbrand, J. Burgers, M. Mäkelä, and J. Slutsky. Improving the quality of health care: using international collaboration to inform guideline programmes by founding the Guidelines International Network (G-I-N). *Quality & safety in health care*, 13:455–460, 2004.

[52] M. Peleg, A. Boxwala, O. Ogunyemi, P. Zeng, S. Tu, R. Lacson, E. Begnstam, and N. Ash. GLIF3: The evolution of a guideline representation format. In *Proceedings of American Medical Informatics Association Symposium*, pages 645–649, 2000.

[53] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E.H. Shortliffe, and M. Stefanelli. Comparing computer-interpretable guideline models: A case-study approach. *Journal of the American Medical Informatics Association*, 10(1):52–68, 2003.

[54] M. Peleg, D. Wang, A. Fodor, S. Keren, and E. Karnieli. *Computer-based medical guidelines and protocols: A primer and current trends*, chapter Adaptation of practice guidelines for clinical decision support: a case study of diabetic foot care. IOS Press, 2008.

[55] R.C. Rasmussen, M.A. Schermann, J.M. Shutske, and Olson DK. Use of the North American guidelines

for children's agricultural tasks with Hmong farm families. *Journal of agricultural safety and health*, 9:265–274, 2003.

[56] K.S. Reddy. Implementation of international guidelines on hypertension: the Indian experience. *Clinical and experimental hypertension*, 21:693–701, 1999.

[57] E. Rhinehart, D.A. Goldmann, and E.J. ORourke. Adaptation of the centers for disease control guidelines for the prevention of nosocomial infection in a pediatric intensive care unit in Jakarta, Indonesia. *The American journal of medicine*, 91:213S–220S, 1991.

[58] D. Riaño, J.A. López-Vallverdú, and S.W. Tu. Mining hospital data to learn SDA* clinical algorithms. In *11th Conference on Artificial Intelligence in Medicine, From Knowledge to Global Care Workshop*, Amsterdam, 2007.

[59] A. Seyfang, R. Kosara, and S. Misch. Asbru's reference manual, asbru version 7.3. Technical Report Asgaard-TR-20002-1, Vienna University of Technology, Institute of Software Technology, 2002.

[60] Y. Shahar, S. Miksch, and P. Johnson. An intention-based language for representing clinical guidelines. In *Proceedings American Medical Informatics Annual Symposium*, pages 592–596, 1996.

[61] Y. Shahar, S. Miksch, and P. Johnson. The asgaard project: A task-specific framework for the application and critiquing of time-orientied clinical guidelines. *Artificial Intelligence in Medicine*, 14:29–51, 1998.

[62] T.M. Shaneyfelt, M.F. Mayo-Smith, and J. Rothwangel. Are guidelines following guidelines? the methodological quality of clinical practice guidelines in the peer-reviewed medical literature. *The Journal of the American Medical Association*, 281:1900–1905, 1999.

[63] P. Shekelle, M.P. Eccles, J.M. Grimshaw, and S.H. Woolf. When should clinical guidelines be updated? *BMJ*, 323:155–157, 2001.

[64] R.N. Shiffman. Towards effective implementation of a pediatric asthma guideline: integration of decision support and clinical workflow support. In *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, pages 791–801, 1994.

[65] J. Shoultz, N. Phillion, J. Noone, and B. Tanner. Listening to women: culturally tailoring the violence prevention guidelines from the put prevention into practice program. *Journal of the American Academy of Nurse Practitioners*, 14:307–315, 2002.

[66] J. Shoultz, B. Tanner, and R. Harrigan. Culturally appropriate guidelines for alcohol and drug abuse prevention. *The Nurse practitioner*, 25:50–56, 2000.

[67] D. Shye, A. Porath, and J.B. Brown. Adapting a national guideline for local use. a comparative case study in a US and an Israeli health maintenance organization. *Journal of health services research & policy*, 5:148–155, 2000.

[68] C.A. Silagy, D.P. Weller, H. Lapsley, P. Middleton, Shelby-James T., and B. Fazekas. The effectiveness of local adaptation of nationally produced clinical practice guidelines. *Family Practice*, 19:123–230, 2002.

[69] E. Sommer. Learning relations without closing the world. In *Proceedings of the European Conference on Machine Learning*, pages 419–422, Berlin, 1994. Springer-Verlag.

[70] E. Sommer. *Theory Restructuring: A Perspective on Design & Maintenance of Knowledge Based Systems*. PhD thesis, University of Dortmund, 1996.

[71] E. Sommer, K. Morik, J. Andre, and M. Usynski. What on-line learning can do for knowledge acquisition. *Knowledge Acquisition*, 6:435–460, 1994.

[72] W.M. Tierney, J.M. Overhage, B.Y. Takesue, L.E. Harris, M.D. Murray, D.L. Vargo, and C.J. McDonald. Computerizing guidelines to improve care and patient outcomes: the example of heart failure. *Journal of the American Medical Informatics Association*, 5:316–322, 1995.

[73] J. Tomlinson, R. Baker, K. Stevenson, and M. Lakhani. The leicestershire local guideline adaptation and implementation programme: a model for clinical effectiveness? *Journal of evaluation in clinical practice*, 6:125–229, 2000.

[74] S. Tu and M. Musen. A flexible approach to guideline modeling. In *Proceedings of American Medical Informatics Association Symposium*, pages 420–424, 1999.

[75] S. Tu and M. Musen. From guideline modeling to guideline execution: Defining guideline based decision-support services. In *Proceedings of American Medical Informatics Association Symposium*, pages 863–867, Los Angeles, CA, 1999.

[76] R. Voellinger, A. Berney, P. Baumann, J.M. Annoni, C. Bryois, T. Buclin, C. Büla, V. Camus, L. Christin, J. Cornuz, P. de Goumoëns, O. Lamy, J. Strnad, B. Burnand, and F. Stiefel. Major depressive disorder in the general hospital: adaptation of clinical practice guidelines. *General Hospital Psychiatry*, 25:185–193, 2003.

[77] J. Wils, P. O'Dwyer, and R. Labianca. Adjuvant treatment of colorectal cancer at the turn of the century: European and US perspectives. *Annals of oncology: official journal of the European Society for Medical Oncology*, 12:13–22, 2001.

[78] S.H. Woolf. Evidence-based medicine and practice guidelines: an overview. *Cancer Control: journal of the Moffitt Cancer Center*, 7(4):362–367, 2000.

[79] S.H. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw. Clinical guidelines: potential benefits, limitations, and harms of clinical guidelines. *BMJ*, 318:527–530, 1999.

[80] S. Wrobel. Automatic representation adjustment in an observational discovery system. In *Proceedings of the Third European Working Session on Learning*, pages 253–262, 1988.

[81] S. Wrobel. First order theory refinement. In *Advances in Inductive Logic Programming*, pages 14–33, Amsterdam, 1996. IOS Press.

[82] O. Ying-Lie. Model-based guideline development for symptom-based indications. In *ECAI 2006 Workshop – AI techniques in healthcare:evidence based guidelines and protocols*, pages 46–51, 2006.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-140

Chapter 8
Visualization Methods to Support Guideline-Based Care Management

Wolfgang AIGNER [a], Katharina KAISER [b], Silvia MIKSCH [a,b,1]
[a] *Danube University Krems, Austria*
[b] *Vienna University of Technology, Austria*

Abstract. Authoring clinical guidelines as well as observing the execution and the maintenance of these is a time-consuming and cumbersome task. Usually, clinical guidelines are represented in conceptual models, which are very hard to understand by domain experts. Furthermore, to analyze the effectiveness and usefulness of clinical guidelines they need to be shown in connection with the patients' data. In this overview book chapter we present different methods to visualize clinical guidelines, patients' data, and the connection thereof. Finally, we illustrate how the different visualization methods support the various tasks in plan management.

Keywords. Visualization, Guideline-based Care

Introduction

Current research into the use of Information Visualization in guideline-based care focuses on support for the knowledge acquisition process (the authoring of computer-interpretable guidelines and protocols) and on ways to help explore plans and monitor and communicate plan execution (over time). In the following we define the concepts of Information Visualization (InfoVis) and the core tasks of plan management to illustrate how InfoVis can support these processes and tasks.

Information Visualization

Growing use of modern information technologies in clinical care is increasing both the amount and complexity of information and data accessible to health professionals. By providing interactive visual representations of data and information, Information Visualization aims to deepen exploration of the "information space", support optimal use of data and information, and help avoid overload. InfoVis is concerned with the development of interactive visual representations of abstract, multidimensional data, information, and knowledge to help users gain a deeper understanding of the contents of a domain

[1] Corresponding Author: Danube University Krems, Department of Information and Knowledge Engineering (ike), Dr.-Karl-Dorrek-Strasse 30, A-3500 Krems, Austria; E-mail: silvia.miksch@donau-uni.ac.at.

by revealing, for example, new insights, previously unknown facts and relationships, or providing explanations for complex situations [4,34].

Chittaro [5] summarizes some of the goals of InfoVis technologies for healthcare:

1. To allow "*users to explore available data at various levels of abstraction*".
2. To give "*users a greater sense of engagement with data*".
3. To give "*users a deeper understanding of data*".
4. To encourage "*the discovery of details and relations which would be difficult to notice otherwise*".
5. To support "*the recognition of relevant patterns by exploiting the visual recognition capabilities of users*".

Visualization methods and tools have been used in the medical domain for many years. The majority of applications have been in the field of scientific visualization, for example, 3D volume visualization tasks, X-ray, or computer tomography visualizations. However, tasks involving abstract data (such as patients' data, treatment data, or lab results) or computerized guidelines have not been targeted by InfoVis researchers until quite recently.

Tasks in Plan Management

In the last years different approaches to manage clinical guidelines and protocols were introduced. These approaches range from contributions of the planning community to concepts within the Medical Informatics community [14]. As a possible solution to that diversity the concept of *plan management* was introduced, where clinical guidelines are seen as (time-oriented) plans [14].

Plan management involves more than specifying a problem, generating a possible solution path to reach a goal state from an initial state, and executing this solution path. Plan management includes everything from designing a particular plan or a hierarchy of plans to the real-world execution and evaluation of such plans (compare [18]). Plan management consists of various tasks. These tasks are not performed in a strict sequence (one by one) and can be clustered into groups according to their purposes. Many of these groups overlap in their functionalities. Additionally, we can distinguish tasks, which need to be performed mainly at design time and those which are done mainly during execution time of plans (as shown in Table 1).

According to Table 1 two tasks are mentioned which address the visualization of clinical guidelines explicitly: (1) *plan visualization during design time*, and (2) *plan and data visualization during execution time*. The first refers to authoring of computer-interpretable clinical guidelines, where the main focus lies on the communication of the different clinical guideline components to domain experts. The second one handles the visual representation of clinical guidelines in connection with patients' data.

However, all other tasks in Table 1 can be supported by visual methods, too. *Advanced Plan Editing* and *Domain-Specific Annotations* can be seen as extended authoring of clinical guidelines, which can be eased by visualization methods. *Plan-Scenario Testing* can be seen as off-line execution of clinical guidelines, which can utilize the plan visualization during execution. Furthermore, *Plan Verification and Validation* could indirectly be supported by visual methods, however, these are quite complicated tasks and verbal annotations seem more suitable. All the remaining tasks during execution time

Table 1. Tasks in Plan Management. A list of tasks defined in plan management according to their primary application time [14].

Tasks mostly done at design time	
Plan Generation	starts from an initial state description and creates a path of activities to reach the desired goal (progressive way), or starting from the goal (regressive way)
Advanced Plan Editing	provides guided support to author plans and helps to browse plans or a plan hierarchy
Domain-Specific Annotations	provides structured support to write domain assumptions or domain activities
Plan Verification	examines the method semantics of plans and plan hierarchies
Plan Validation	examines the domain semantics of plans and plan hierarchies
Plan-Scenario Testing	enacts groups of plans by applying definite domain scenarios
Plan Visualization	communicates efficiently sets of plans to domain experts
Tasks mostly done at execution time	
Plan Selection	assigns task according to state/situation
Plan Adaptation	adjusts plans or plan hierarchies to distinct situations at the starting time
Plan Execution	performs selected activities
Plan Monitoring	compares assumptions with reality
Plan Modification / Alternatives	handles changes in the environment
Plan Evaluation / Critiquing	analyzes executed plans or plan hierarchies according to their goals and intentions
Plan and Data Visualization	supervises and communicates plans or plan hierarchies in connection with the patients' data
Plan Rationale / History	explains executed plans or plan hierarchies

can be supported by visual methods in one way or another. One outstanding task is *Plan Modification / Alternatives*: This covers, on the one hand, the maintaining of clinical guidelines when new medical knowledge is discovered and needs to be incorporated in the clinical guidelines (in the sense of "*living guidelines*"). On the other hand, changes in health condition of the patients or in the medical environment can force a modification of the therapeutic activities.

We structure the visualization methods according to abstractions of these tasks: (1) visualizing clinical (computer-interpretable) guidelines seen as plans or activities, (2) visualizing patients' data seen as multidimensional information space, and (3) visualizing patients' data in connection with clinical guidelines. We illustrate afterwards, how the visualization methods can be utilized in design time as well as execution time (compare Table 2).

1. Visualizing Clinical Guidelines seen as Plans or Activities

Different frameworks have been developed to implement clinical guidelines in a computer-interpretable format, such as Asbru, EON, GLIF, Guide, Prodigy, and PROforma (compare [16,7] and Chapter 2 in this book). These frameworks are tailored for specific classes of guidelines, specific users, and specific organizations. Each framework supports specific guideline representation languages and various tools and techniques have been developed to ease the guideline modeling and visualization process. They can be roughly classified into (1) *model-centric* and (2) *document-centric* approaches [13]. In the model-centric approach, a conceptual model is formulated by domain experts. Thus, the relationship between the model and the original document of the clinical guideline is only indirect. In the document-centric approach, markup-based tools are used to systematically mark up the original guideline in order to generate a semi-formal model of the marked text part. The first category covers many approaches and is more visual-oriented, the latter is more text-based and only a few approaches are available. In the following we present a selection of examples according to these two categories. There are different flowchart-based visualizations of clinical guidelines available and Protégé (compare Section 1.1.3) is used in different frameworks. We only present representatives of these.

1.1. Model-Centric Approaches

The *model-centric* approaches to author clinical guidelines focus on the creation of a conceptual model of the original guideline without keeping the direct connection between these two representations. We illustrate this approach by various examples ranging from simple flowcharts to sophisticated visual representations of guidelines.

1.1.1. Clinical Algorithm Maps

The most widely used visual representation of clinical guidelines are so-called *flowchart algorithms*, also known as *clinical algorithm maps* [9]. A standard for this kind of representation has been proposed by the *Committee on Standardization of Clinical Algorithms* of the *Society for Medical Decision Making* [27]: *"However, since algorithmic logic is wired implicitly into a protocol, it is difficult to learn an algorithm from a protocol. By contrast, flowchart algorithms, or clinical algorithm maps, are uniquely suited for explicitly communicating conditional logic and have therefore become the main format for representing a clinical algorithm clearly and succinctly."* The proposed standard includes a small number of different symbols and some rules on how to use them. One additional feature to standard flowcharts are annotations that include further details, e.g., citations to supporting literature, or clarifications for the rationale of decisions.

A big advantage of using flowcharts is that they are well known among physicians and require minimal additional learning effort. A drawback of basic flowchart representations is their immense space consumption if more complex situations are depicted where overview is lost easily. Temporal information can only be represented implicitly on a very coarse level in terms of an item's relative position within a sequence (before, after). Furthermore, flowcharts cannot be used to represent concurrent tasks or the complex conditions as used in Asbru [24] due to their state-like semantics. Clinical algorithm maps were intended to be used on paper and have never been enriched by computer support, such as navigation or versatile annotation possibilities.

1.1.2. Nassi-Shneiderman Diagrams, PERT Charts, Gantt Charts, Petri Nets, and State Transition Diagrams

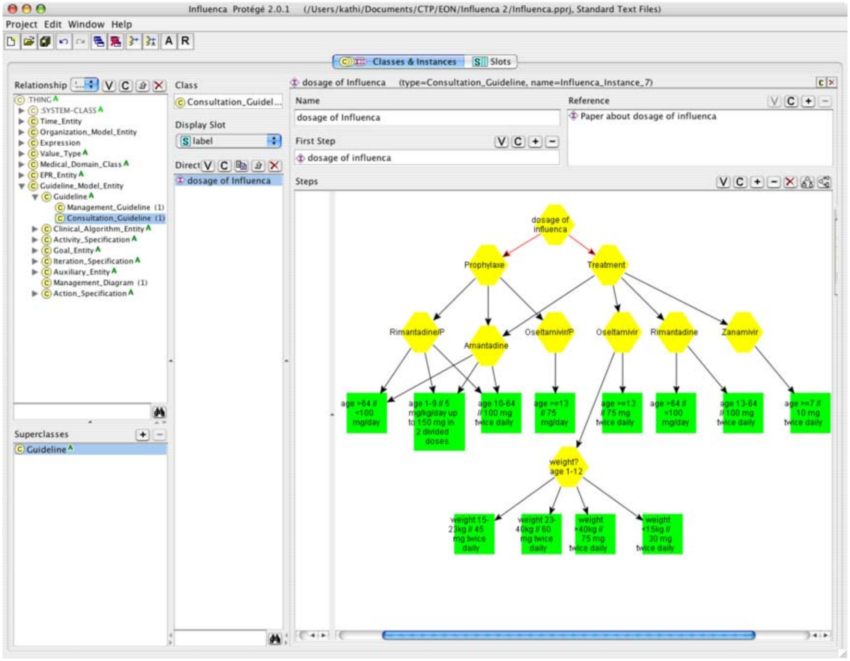
Other possibilities to visualize logical sequences besides flowcharts are *Structograms (Nassi-Shneiderman Diagrams)*, *PERT charts*, *Gantt charts*, *Petri nets*, and *State Transition Diagrams*. These techniques focus on other purposes and some of them are more powerful and expressive than flowcharts. But none of them offers a notion for a combined depiction of hierarchical decomposition, flexible execution order, and the state characteristic of conditions in their basic forms as needed for representing clinical guidelines.

1.1.3. Protégé

Protégé is an open source ontology development and knowledge acquisition environment [8]. Protégé is a Java tool, which provides an extensible architecture for the creation of customized knowledge-based tools. It assists users in the construction of large electronic knowledge bases. It has an intuitive user interface that enables developers to create and edit domain ontologies and supports customized user-interface extensions, incorporates the Open Knowledge Base Connectivity (OKBC) knowledge model, and interacts with standard storage formats such as relational databases, XML, and RDF. Protégé supports to author guidelines in various models (e.g., EON, GLIF, Prodigy, Proforma). See Figure 1 for an example.

1.1.4. VisiGuide (Part of DeGeL)

The VisiGuide is part of the DeGeL (Digital electronic Guideline Library) project [25] and is a multi-ontology guidelines browser. Its purpose is to present a large amount of guidelines clustered by the semantic indices and allow the user focusing on a single guideline by exploring its parts (see Figure 2).


Influenca Protégé 2.0.1 (/Users/kathi/Documents/CTP/EON/Influenca 2/Influenca.pprj, Standard Text Files)

Class: Consultation_Guidel...

Display Slot: label

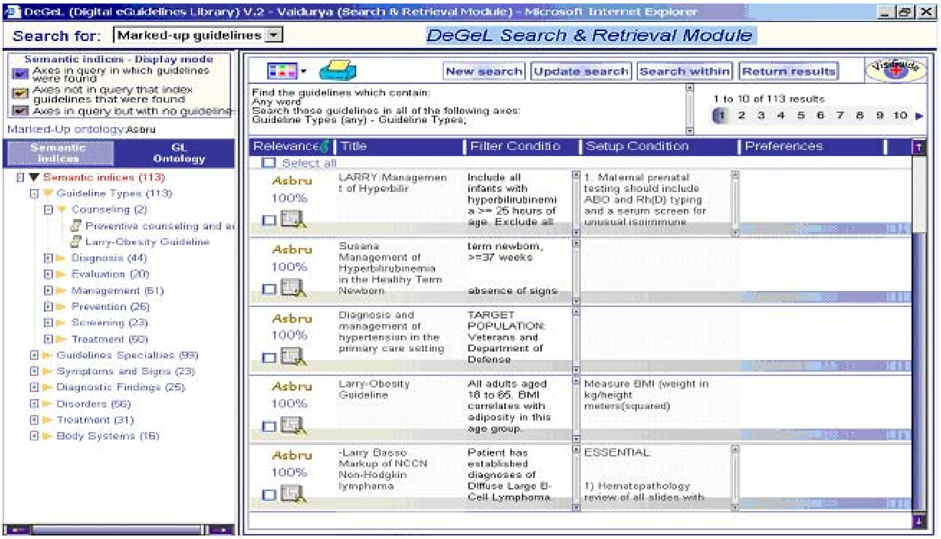
dosage of Influenca (type=Consultation_Guideline, name=Influenca_Instance_7)

Name: dosage of Influenca

First Step: dosage of influenca

Reference: Paper about dosage of influenca

Figure 1. Protégé – A knowledge acquisition tool to author a guideline for managing chronic cough. The guideline model being used in this application is Dharma, part of the EON framework.



Find the guidelines which contain:
Any word
Search these guidelines in all of the following axes:
Guideline Types (any) - Guideline Types;

Relevance	Title	Filter Conditio	Setup Condition	Preferences
□ Asbru 100%	LARRY Managemen t of Hyperbilir	Include all infants with hyperbilirubinemi a >= 25 hours of age. Exclude all	1. Maternal prenatal testing should include ABO and Rh(D) typing and a serum screen for unusual isoimmune	
□ Asbru 100%	Susana Management of Hyperbilirubinemia in the Healthy Term Newborn	term newborn, >=37 weeks absence of signs		
□ Asbru 100%	Diagnosis and management of hypertension in the primary care setting	TARGET POPULATION: Veterans and Department of Defense		
□ Asbru 100%	Larry-Obesity Guideline	All adults aged 18 to 65. BMI correlates with adiposity in this age group.	Measure BMI (weight in kg/height meters(squared)	
□ Asbru 100%	-Larry Basso Markup of NCCN Non-Hodgkin lymphoma	Patient has established diagnoses of Diffuse Large B-Cell Lymphoma	ESSENTIAL: 1) Hematopathology review of all slides with	

Figure 2. VisiGuide [25] – A visualization and browsing tool for multiple guidelines.

1.1.5. AsbruView - SopoView

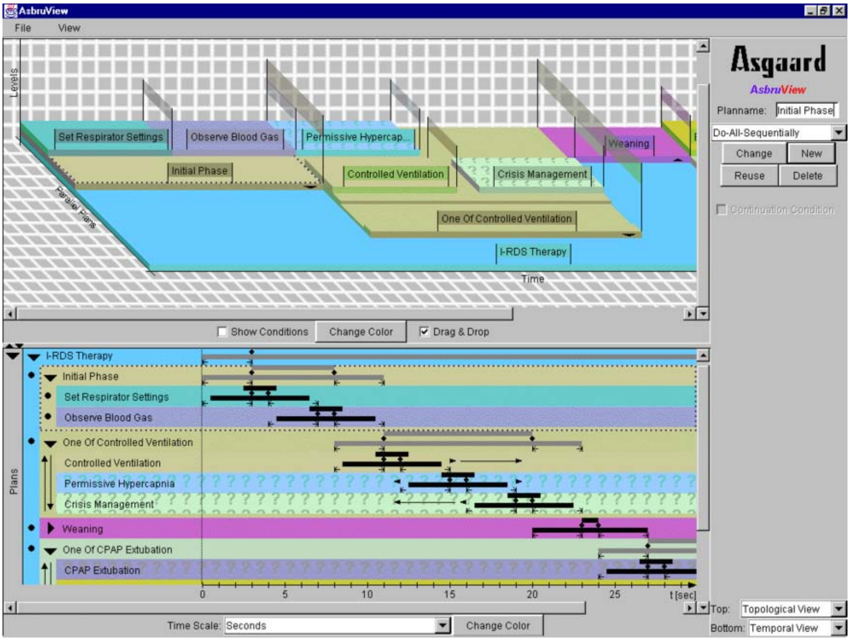
AsbruView [12][2] is a graphical user interface developed in the Asgaard/Asbru project to support the development of guidelines and protocols in Asbru [24]. Asbru is a complex language, which cannot be fully understood by physicians who have no or hardly any training in formal methods. AsbruView is a tool to make Asbru accessible to physicians, and to provide a visual overview of the guideline hierarchy and other Asbru-specific components. AsbruView is based on visual metaphors of running tracks and traffic signs to make the underlying concepts easier to grasp. Currently, AsbruView (see Figure 3) provides four views: *Topological View* (see upper part of Figure 3) displays the relationship between guidelines seen as plans without a precise time scale, *Temporal View* (see lower part of Figure 3) concentrates on the temporal dimensions of plans and conditions, *SOPOView* gives another view of the temporal dimensions of plans, and *XML View*. The metaphors and graphical representations of *AsbruView* have proved to be useful in communicating Asbru's concepts to physicians. Users get a better overview of the therapy steps than from tables, while at the same time being able to see the precise temporal constraints of plans (which is not the case with flowcharts).

1.1.6. AsbruFlow (Part of CareVis)

AsbruFlow[3] is part of CareVis prototype [2] and helps to communicate the content and logic of Asbru treatment plans to medical domain experts [24]. AsbruFlow is based on the idea of flowchart-like clinical algorithm maps [9] that are well known amongst physicians. This concept has been extended in order to be able to depict the characteristics of a treatment plan formulated in Asbru.

[2] http://www.asgaard.tuwien.ac.at/tools/asbruview.html [accessed May 2007]

[3] http://ieg.ifs.tuwien.ac.at/projects/carevis/ [accessed June 2007]



Planname: Initial Phase

Do-All-Sequentially

Time Scale: Seconds

Top: Topological View

Bottom: Temporal View

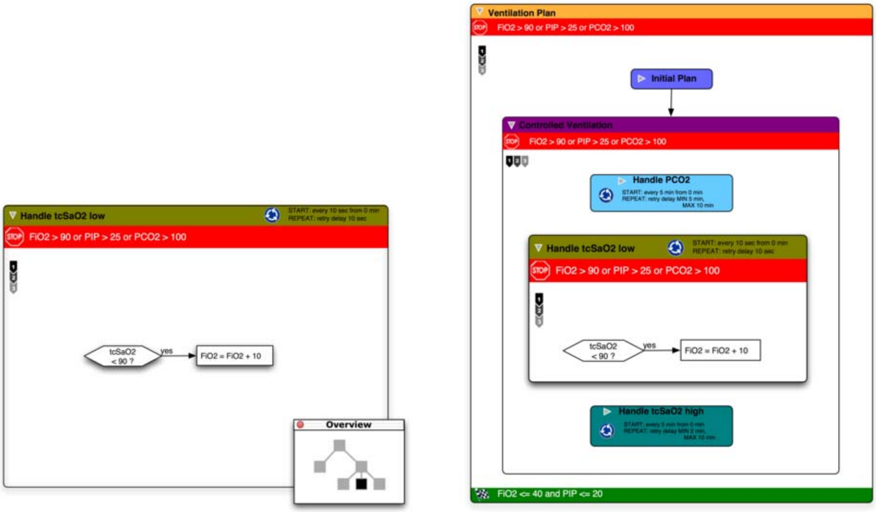
Figure 3. AsbruView [12] – A graphical user interface for the guideline representation language Asbru showing a part of a guideline for infants' respiratory distress syndrome (I-RDS) of new-born infants. *Topological View* is in the upper part and *Temporal View* in the lower part [12].

In order to prevent getting lost within a guideline by navigation, two *Focus+Context* techniques are applied. Firstly, there is the *Overview+Detail* mode that uses a small window containing a downscaled, simplified tree overview where the current position within the plan is highlighted. This small overview window can be toggled on or off (see Figure 4, left (a)). The second technique is a *Fisheye view* which distorts elements that are out of the current focus geometrically by shrinking and moving (see Figure 4, right (b)) based on the work of Schaffer et al. on hierarchically clustered networks [22].

1.1.7. Arezzo – Tallis Toolset

The first implementation of software to create, visualize, and enact PROforma guidelines [29] was Arezzo. Its successor is the Tallis Toolset [28].[4] It includes software and training materials to create, publish, and enact clinical knowledge applications over the web. It is based on the PROforma language [29] for modeling clinical processes. The toolset consists of three components that interact with each other: the Composer, the Tester, and the Engine. PROforma tasks (i.e., plans, decisions, actions, enquiries) can be connected to form a network. Such a network is sometimes called a "workflow". The Composer provides a graphical interface to support the generation of such task networks. The development of a network is a two-step process: (1) a high-level structure of the process is laid out and assembled as a network; (2) detailed knowledge that is required to enact each component task is entered as task attributes. The Tallis Tester is a tool for testing

[4] http://www.acl.icnet.uk/lab/tallis [accessed May 2007]


Handle tcSaO2 low
START: every 10 sec from 0 min
REPEAT: retry delay 10 sec
FiO2 > 90 or PIP > 25 or PCO2 > 100
tcSaO2 < 90 ?
yes
FiO2 = FiO2 + 10
Overview
Ventilation Plan
FiO2 > 90 or PIP > 25 or PCO2 > 100
Initial Plan
Controlled Ventilation
FiO2 > 90 or PIP > 25 or PCO2 > 100
Handle PCO2
START: every 5 min from 0 min
REPEAT: retry delay MIN 5 min, MAX 10 min
Handle tcSaO2 low
START: every 10 sec from 0 min
REPEAT: retry delay 10 sec
FiO2 > 90 or PIP > 25 or PCO2 > 100
tcSaO2 < 90 ?
yes
FiO2 = FiO2 + 10
Handle tcSaO2 high
START: every 5 min from 0 min
REPEAT: retry delay MIN 2 min, MAX 10 min
FiO2 <= 40 and PIP <= 20

(a) Overview+Detail.

(b) Fisheye view.

Figure 4. AsbruFlow showing parts of the Asbru plan for artificial ventilation of newborn infants – Overview+Detail mode (left (a)) vs. Fisheye view mode (right (b)).

and debugging the logic of a developed PROforma application. A tested and debugged application can then be enacted by the Tallis Engine.

1.1.8. GLARE

GLARE [30] is a domain-independent system for acquiring, representing, and executing clinical guidelines. The graphical interface of the authoring tool is quite similar to the Tallis Toolset and the applications of Protégé. The different language elements are coded by simple graphical icons and the flow of the guideline is represented similar to a flowchart.

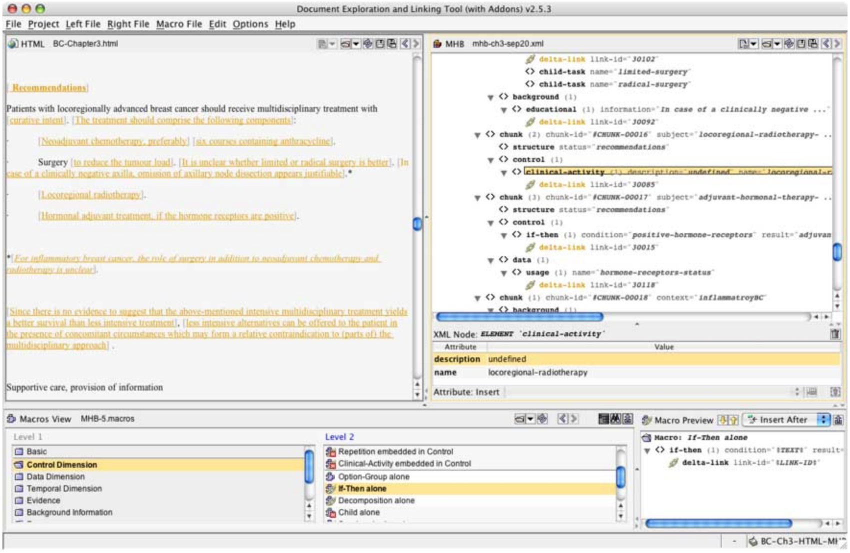
1.2. Document-Centric Approaches

The *document-centric* approach to author clinical guidelines preserves the connection of the original guideline written in text and its semi-formal model. This approach is more text-based and only a few examples exist. To illustrate this approach we give two examples, next. Similar markup tools are provided in the Stepper project [21] and the Uruz project, which is part of the DeGeL project [25].

1.2.1. GEM Cutter

The GEM Cutter [19][5] facilitates the transformation of clinical guidelines into the Guideline Elements Model (GEM) [26], which is an XML-based guideline document model. GEM Cutter's main screen consists of three vertical segments showing the original text

[5] http://gem.med.yale.edu/GEMCutter/gemcutter.htm [accessed: June 2007]



[Recommendations]

Patients with locoregionally advanced breast cancer should receive multidisciplinary treatment with [curative intent]. [The treatment should comprise the following components]:

- [Neoadjuvant chemotherapy, preferably] [six courses containing anthracycline].
- Surgery [to reduce the tumour load]. [It is unclear whether limited or radical surgery is better]. [In case of a clinically negative axilla, omission of axillary node dissection appears justifiable].*
- [Locoregional radiotherapy].
- [Hormonal adjuvant treatment, if the hormone receptors are positive].

*[*For inflammatory breast cancer, the role of surgery in addition to neoadjuvant chemotherapy and radiotherapy is unclear*].

[Since there is no evidence to suggest that the above-mentioned intensive multidisciplinary treatment yields a better survival than less intensive treatment], [less intensive alternatives can be offered to the patient in the presence of concomitant circumstances which may form a relative contraindication to (parts of) the multidisciplinary approach] .

Supportive care, provision of information

```
            delta-link link-id="30102"
            <> child-task name="limited-surgery"
            <> child-task name="radical-surgery"
      ▼ <> background (1)
         ▼ <> educational (1) information="In case of a clinically negative ..."
            delta-link link-id="30092"
   ▼ <> chunk (2) chunk-id="#CHUNK-00016" subject="locoregional-radiotherapy- ..
      <> structure status="recommendations"
      ▼ <> control (1)
         ▼ <> clinical-activity (1) description="undefined" name="locoregional-r
            delta-link link-id="30085"
   ▼ <> chunk (3) chunk-id="#CHUNK-00017" subject="adjuvant-hormonal-therapy- ..
      <> structure status="recommendations"
      ▼ <> control (1)
         ▼ <> if-then (1) condition="positive-hormone-receptors" result="adjuvan
            delta-link link-id="30015"
      ▼ <> data (1)
         ▼ <> usage (1) name="hormone-receptors-status"
            delta-link link-id="30118"
   ▼ <> chunk (1) chunk-id="#CHUNK-00018" context="inflammatroyBC"
      ▼ <> background (1)
```

XML Node: *ELEMENT* 'clinical-activity'

Attribute	Value
description	undefined
name	locoregional-radiotherapy

Attribute: Insert

```
Macro: If-Then alone
▼ <> if-then (1) condition="$TEXT$" result=
      delta-link link-id="$LINK-ID$"
```

Figure 5. Document Exploration and Linking Tool / Addons (DELT/A) – The two top panes show linked views of the same guideline in different formats and the lower pane is the Macro view to ease authoring of guidelines.

of the guideline, a tree view of the developing GEM file, and additional information about the GEM file. It is a pure text-based approach.

1.2.2. Document Exploration and Linking Tool / Addons (DELT/A)

DELT/A [33] [6] provides two main features: (1) linking between a textual guideline and its formal representation, and (2) applying design patterns in the form of macros. DELT/A's user interface (see Figure 5) consists of three panes. The top left and right panes provide equivalent views to either edit XML files or HTML files. The Macros pane provides either a structure view, search view, or insertable macros view, as well as a preview of the current macro.

DELT/A allows the definition of links between the original guideline and the target representation, which gives the user the possibility to find out where a certain value in the XML-language notation comes from. Using macros allows creating and extending specific XML files more easily through the usage of common design patterns.

In this section we have illustrated how clinical guidelines seen as plans and processes can be visualized. In the next section we give examples of visualizing patients' data.

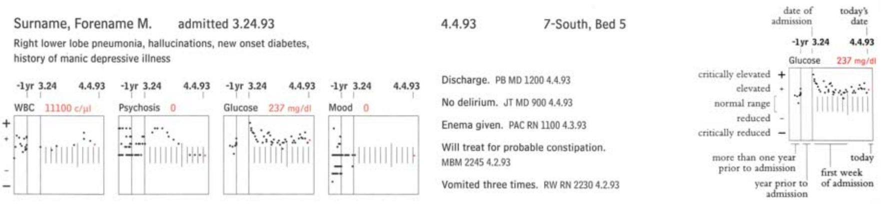
2. Visualizing Patients' Data seen as Multidimensional Information Space

In medicine, large amounts of information are generated and have to be processed mostly by humans. Graphical representations help to make this myriad of information graspable

[6] http://ieg.ifs.tuwien.ac.at/projects/delta/ [accessed May 2007]

and are a crucial part in the workflow of healthcare personnel. These representations range from classical Fever Curves and EEG Time-Series Plots as found in many commercial patient data monitoring systems to information rich patient's status overviews (see Figure 6). The graphical summary of patient's status by Tufte and Powsner [20] makes use of concepts like small multiples, Focus+Context, or the integration of textual and graphical information. It manages to display information on a single page that normally fills up entire file folders and would require serious effort to summarize it.

Conceptually, patients' data can be seen as multidimensional information spaces. These information spaces are heterogeneous in multiple ways – quantitative and qualitative data; a mixture of numerical values, text, and images; high-frequency and low-frequency data; raw data and data abstractions. Particularly, this heterogeneity and the need to provide integrated views to create a comprehensive picture of a patient's status as well as its evolution over time impose substantial challenges to visualization design. Besides the visual representation itself, interactivity is a prime concern. It allows for an active interplay of the user and the visualization in order to adapt to the user's information needs for particular tasks and provide additional detail where needed.



Surname, Forename M. admitted 3.24.93

Right lower lobe pneumonia, hallucinations, new onset diabetes, history of manic depressive illness

4.4.93 7-South, Bed 5

Discharge. PB MD 1200 4.4.93

No delirium. JT MD 900 4.4.93

Enema given. PAC RN 1100 4.3.93

Will treat for probable constipation. MBM 2245 4.2.93

Vomited three times. RW RN 2230 4.2.93

(a) Part of overview screen.

(b) Legend.

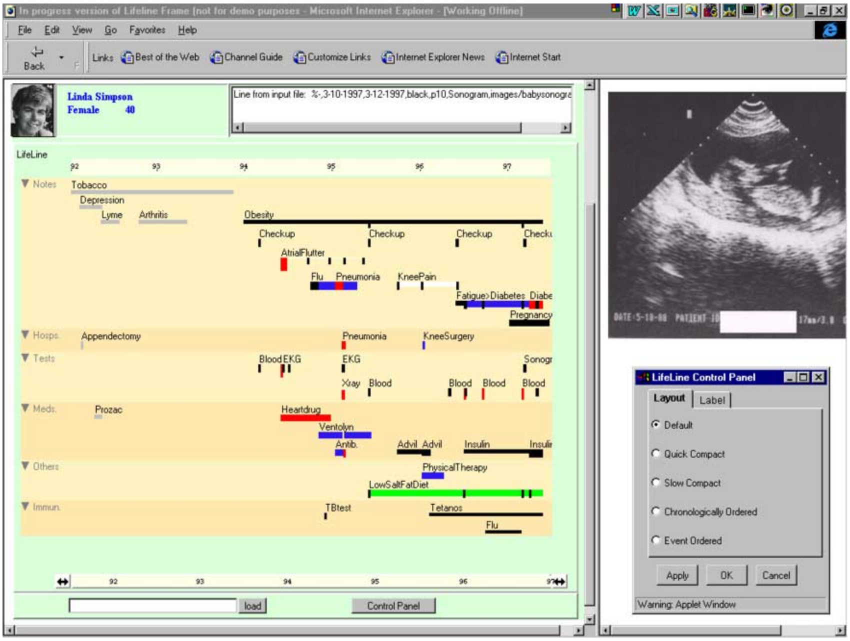
Figure 6. Graphical Summary of Patient's Status [20] – Concise summary of patient's information. Uses *small multiples*, *Focus+Context*, and integrates textual as well as graphical information.

2.1. Data and Information Visualization

In the upcoming section, we present information visualization techniques for patient's data that go beyond simple line plots of patient's data monitoring systems.

2.1.1. Time Lines and LifeLines

A simple and intuitive way of depicting time-oriented activities is by drawing a horizontal line for the time span the activities took. This form of visualization is called *Time Line* which is a very powerful visualization technique used long before computers even appeared [32]. An extension of Time Lines are *LifeLines* [17] which utilize horizontal bars to represent the temporal location and duration of data (see Figure 7). They were applied to represent personal histories and patient's records. In order to organize the elements, so-called "facets" are introduced for grouping the data which can be expanded and collapsed. When collapsed, only a very small and geometrically as well as semantically downscaled version without textual labels is shown. Furthermore, information can be encoded via the height and color of individual bars. Additional information can be provided on demand in a linked view, as, for example, X-ray images.



Line from input file: %-,3-10-1997,3-12-1997,black,p10,Sonogram,images/babysonogra

● Default
○ Quick Compact
○ Slow Compact
○ Chronologically Ordered
○ Event Ordered

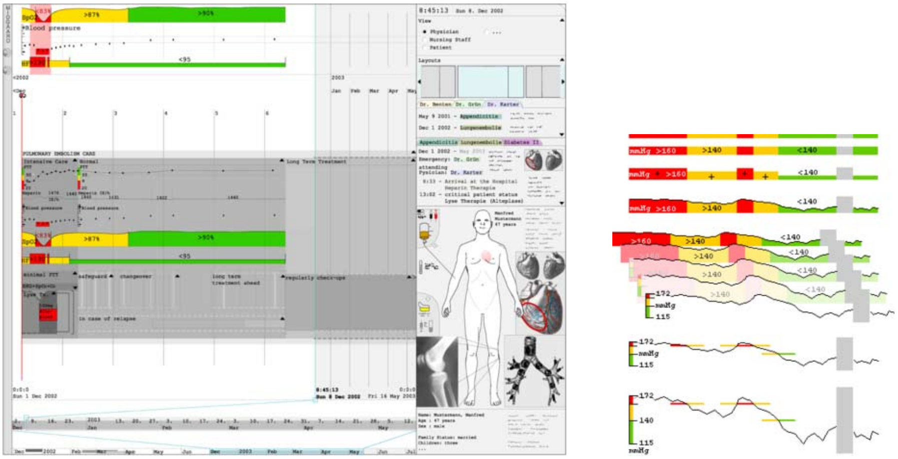
Figure 7. LifeLines [17] – Horizontal bars are used to show the temporal location and duration of information elements. Example shows patient's information.

2.1.2. Midgaard

Midgaard is an interactive and adaptive visualization technique for intensive care data (see Figure 8(a)). It provides a number of user interactions such as browsing, pan+zoom, focus+context, and smoothly integrated semantic zoom functionality (see Figure 8(b)). In addition to the visualization of quantitative time-series data, qualitative scales as well as qualitative/quantitative hybrids might be displayed. Moreover, Midgaard allows for simultaneous representation of high- and low frequency data typically found in medical care through intelligent zooming and aggregation techniques. In the value domain, methods for coping and representing measurement deviation, trustability of data points, and missing data are provided. Furthermore, Midgaard visualizes patients' data in combination with assigned clinical guidelines.

2.1.3. VIE-VISU

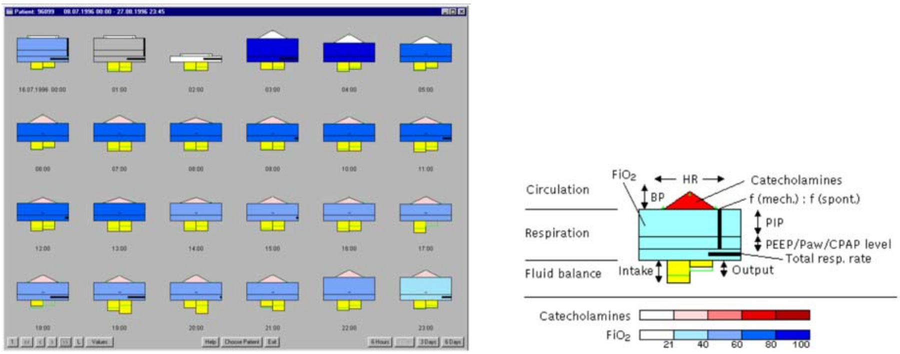
An interactive glyph technique that is used for time-oriented analysis of electronic patient's records is *VIE-VISU* [11]. A glyph is a graphical object using different geometric and visual attributes to encode multidimensional data structures in combination [34]. The motivation for VIE-VISU's development was the fact that paper-based analysis of patient's records is very hard to conduct because many parameters are involved and an overall assessment of the patient's situation is hard to maintain. The glyph display helps to combine different measurements, maintain their relationships, show their development over time, and make specific, possibly life threatening situations, easy to spot. The used glyph basically consists of three parts that represent circulation, respiration, and fluid



(a) Main screen.

(b) Semantic zoom which adapts the type of representation depending on the available screen space.

Figure 8. Midgaard [3] – Visualization of intensive care data. Integrated visualization of quantitative time-series data, qualitative scales as well as qualitative/quantitative hybrids.



(a) VIE-VISU Interface – Each glyph represents a one hour period and 24 glyphs are combined in a single screen.

(b) Glyph Parts – The glyph basically consists of three parts that represent circulation, respiration, and fluid balance parameters (top to bottom. All in all, 15 different patient's parameters are combined to form this glyph.

Figure 9. VIE-VISU [11] – Combines different measurements, maintaining their relationships, shows their development over time, and make specific, possibly life threatening situations, easy to spot. Example shows neonatal patient's data.

balance parameters (top to bottom, see Figure 9(b)). All in all, 15 different patient's parameters are combined to form this glyph. Each glyph represents a one hour period and 24 glyphs are combined in a single screen (see Figure 9(a)). Different time frames can be selected and displayed (from 6 hours to 6 days).

2.1.4. Interactive Parallel Bar Charts (IPBC)

Interactive Parallel Bar Charts (IPBC) [6] have been designed to analyze hemodialysis data. It uses rows of histogram cuboids to represent the data in 3D (see Figure 10). In terms of interacting with the technique, several useful features, as, for example, the "tide mode" that allows for highlighting specific volumes and brushing of the representation, are included. Color mappings can be changed interactively and pattern matching can be performed based on examples not only for exact matches, but also with configurable tolerance levels. Moreover, the problem of occlusions in the 3D representation is addressed by providing the possibility to flatten occluding elements. All interactions are smoothly integrated by animated transitions when changing view parameters or interacting with the representation.

2.1.5. Gravi++

A visual method to analyze data gathered from patients via questionnaires is *Gravi++* [10]. Gravi++ was designed to find predictors during the treatment planning for anorectic girls. It utilizes human capabilities by positioning icons on the screen: icons representing patients and questions from the questionnaires. This is modeled with a spring-based system where every question is connected with every person by an (invisible) spring. Every person's icon position on the screen identifies how she answered the respective questions. This leads to the formation of clusters of persons who gave similar answers (see Figure 11). To visualize the changing values over time, Gravi++ uses animation whereas the position of each person's icon changes over time allowing to trace, compare, and analyze the changing values. Alternatively, the change over time can also be represented by traces.

2.2. Data Abstraction

In the previous section, we have presented a number of interactive techniques for visualizing and analyzing patients' data. In the course of studying and analyzing patients' data, users are mostly interested in the meaning that can be derived from the raw data, that is, the *information* that can be extracted. For example, the exact reading of the body temperature of a patient might not be as important as the fact whether the patient has "low fever" or "high fever". Additionally, the abstraction of raw-data to cognitively higher concepts yields data reduction of often huge amounts of data. This data reduction, in turn, eases visualization and analysis tasks. The objective of data abstraction in general is *"to create an abstraction that conveys key ideas while suppressing irrelevant details"* [31].

Examples for such systems used for temporal data abstraction (TDA) are *KNAVE II* and *VIE-VENT*. KNAVE II [23] is a tool that supports the visualization, summarizing, (intelligent) interpretation, explanation, and context-sensitive navigation of time-oriented raw clinical data sets and higher-level concepts abstracted from time-oriented data. *VIE-VENT* [15] addresses context-sensitive and expectation-guided temporal abstraction methods in a medical application domain. The developed methods incorporate knowledge about data points, data intervals, and expected qualitative trend patterns to arrive at unified qualitative descriptions. Smoothing and adjustment mechanisms are used to keep qualitative descriptions stable in case of shifting contexts or data oscillating near thresholds. For example, during intermittent positive pressure ventilation (IPPV),

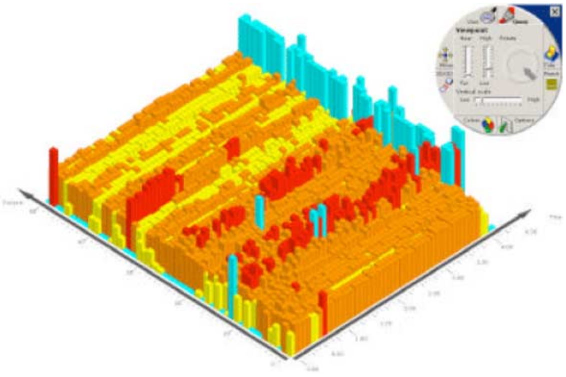


Figure 10. Interactive Parallel Bar Charts (IPBC) [6] – Rows of histogram cuboids for each series of measurements represent the data in 3D.

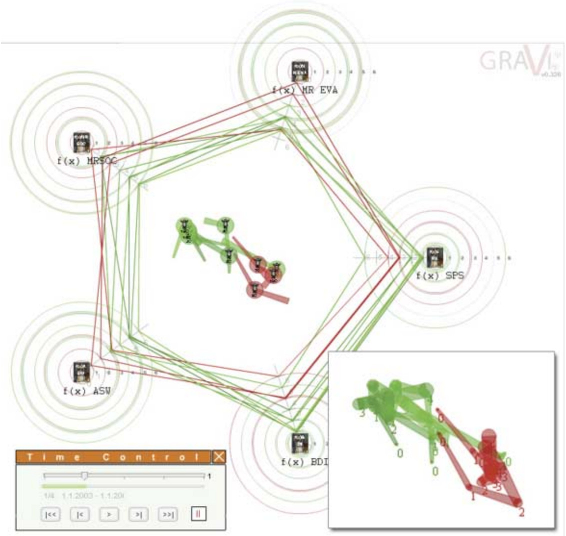


Figure 11. Gravi++ [10] – Patient's icons in the middle of the display are positioned according to a spring-based model relative to the surrounding question icons. The representation might be stepped through time manually or animated via the time control panel on the lower left. Furthermore, traces might be displayed that convey information about the evolution of values over time as shown in detail on the lower right.

the transformation of the quantitative value $P_{tc}CO_2$ = $56mmHg$ results in a qualitative $P_{tc}CO_2$ value of "substantially above target range".[7] During intermittent mandatory ventilation (IMV), however, $56mmHg$ represents the "target value".

In the previous two sections we have shown how guidelines and patients' data can be visualized separately. In the next section we depict how these two views can be connected with in one.

[7] $P_{tc}CO_2$ = transcutaneous partial pressure of carbon dioxide

3. Visualizing Patient Data in Connection with Clinical Guidelines

Visualizing clinical guidelines in combination with patients' data is a challenging task, because heterogeneous and time-oriented data and information need to be visualized in an intuitive way. Usually, the visual representations of clinical guidelines are simply combined with text-based explanations of the course of the patient. However, very few contributions exist, which try to propose more sophisticated solutions. In the following we describe three representatives.

3.1. Tallis Tester

In Section 1.1.7 we explained the Arezzo and Tallis tools to model PROforma guidelines. One component of the Tallis Toolset is the Tallis Tester, which supports the enacting of guidelines. The Tallis Tester keeps track of which tasks need to be performed and provides information to external data and events regarding the current state of the execution.

3.2. CareVis

CareVis is an interactive visualization method to support the visualization of Asbru's plan execution and monitoring [2]. CareVis provides multiple simultaneous views to cover different aspects of a complex underlying data structure of treatment plans and patient's data.

Basically, CareVis divides the underlying data structure along the lines of logical structure and temporal aspects. Hence, CareVis provides a *Logical View* (compare *AsbruFlow* in Section 1.1.6) and a *Temporal View* along with a *QuickView panel*. These distinct views are presented simultaneously and divide the screen in the following manner (see Figure 12). The QuickView panel is located on top of the screen displaying the most important patient's parameters and plan variables at a distinct position. Below that, the screen is divided vertically by the logical view on the left and the temporal view on the right-hand side. The logical view presents treatment plans in terms of their logical structure (hierarchical decomposition, plan elements, execution order, conditions). The temporal view, on the other side, focuses on the temporal aspects of treatment plans and measured patient data as well as plan variables (temporal aspects of plan elements, temporal uncertainties, hierarchical decomposition).

3.3. Guideline Overview Tool - GOT

The main purpose of the *Guideline Overview Tool (GOT)* [1] is to provide a compact overview that is easy to read and comprehend in order to support physicians in executing and analyzing therapies with clinical guidelines (see Figure 13). GOT's overview has two main functions: (1) showing the actual state of a patient in relation to the assigned clinical guideline and (2) displaying several patients at one view in order to compare them.

In the next section, we illustrate how the different visualization methods facilitate the tasks in plan management.

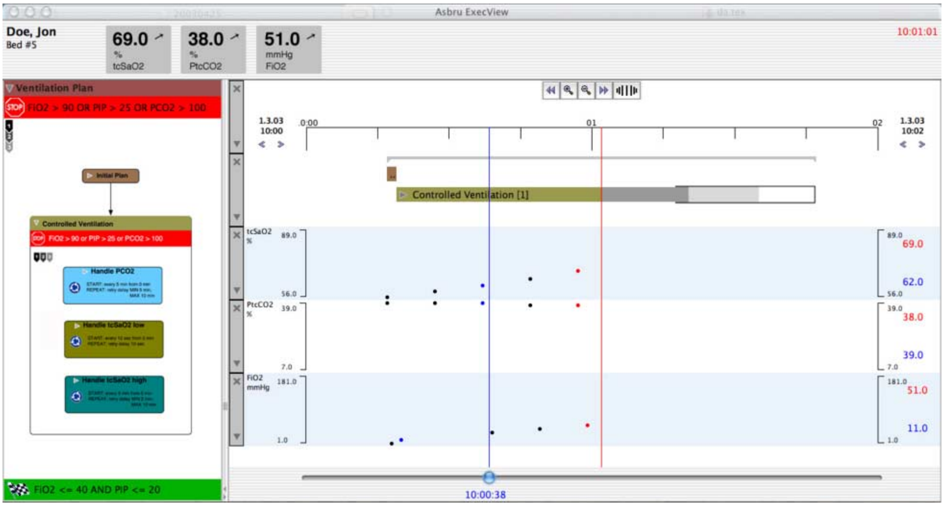
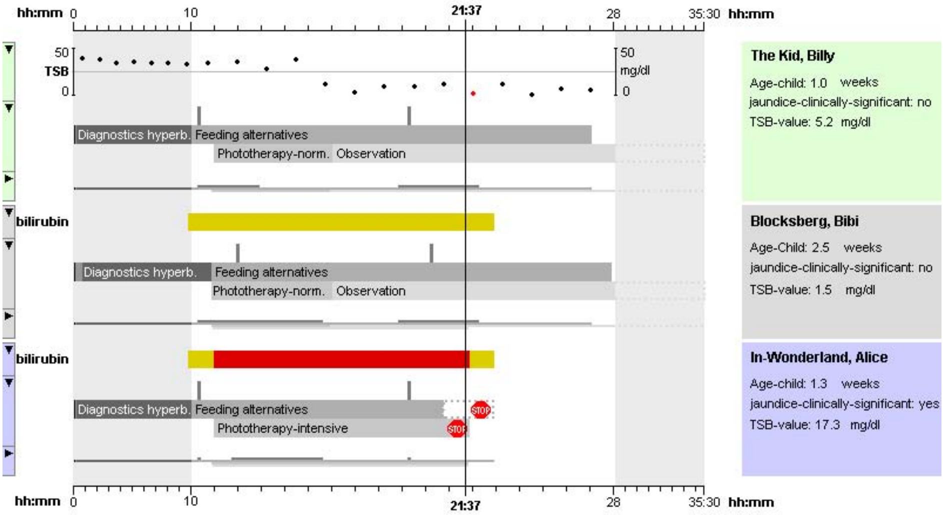


Figure 12. CareVis prototype [2] – Showing *Logical View* on the left-hand side, *Temporal View* on the right–hand side, and *QuickView panel* above.



The Kid, Billy
Age-child: 1.0 weeks
jaundice-clinically-significant: no
TSB-value: 5.2 mg/dl

Blocksberg, Bibi
Age-Child: 2.5 weeks
jaundice-clinically-significant: no
TSB-value: 1.5 mg/dl

In-Wonderland, Alice
Age-child: 1.3 weeks
jaundice-clinically-significant: yes
TSB-value: 17.3 mg/dl

Figure 13. GOT [1] – It shows different patients treated with the same guideline for Hyperbilirubinemia.

4. Visual Supporting Tasks

The main tasks of plan management were discussed in the first section (compare Table 1). Afterwards we gave an overview about the different visualization methods. Table 2 sketches how these tasks can be best supported by the different visualization methods.

Table 2. Visualization Methods and Tools Used for Tasks in Plan Management.

Tasks mostly done at design time	**Visualization methods and tools**
Plan Generation	*only for visualizing the output*
Advanced Plan Editing	*Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, GLARE, GEM Cutter, DELT/A*
Domain-Specific Annotations	*Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, GLARE, GEM Cutter, DELT/A*
Plan Verification	*only for visualizing the output*
Plan Validation	*only for visualizing the output*
Plan-Scenario Testing	*Protégé, AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*
Plan Visualization	*clinical algorithm maps, Nassi-Shneiderman diagrams, PERT charts, Gantt charts, and Petri nets, Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, GLARE, Tallis Tester, CareVis, GOT*

Tasks mostly done at execution time	**Visualization methods and tools**
Plan Selection	*clinical algorithm maps, Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Tester, GLARE, CareVis*
Plan Adaptation	*clinical algorithm maps, Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, DELT/A, Tallis Tester, GLARE, CareVis, GOT*
Plan Execution	*AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*
Plan Monitoring	*AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*
Plan Modification / Alternatives	*Protégé, VisiGuide, AsbruView, SopoView, AsbruFlow, Tallis Composer, GLARE, GEM Cutter, DELT/A*
Plan Evaluation / Critiquing	*AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*
Plan Visualization	*Protégé, GLARE, Tallis Tester, CareVis, GOT*
Data Visualization	*graphical summary of patient's status by Tufte/Powsner, Time Lines, LifeLines, Midgaard, VIE-VISU, IPBC, Gravi++, VIE-VENT, KNAVE II, Tallis Tester, CareVis, GOT*
Plan and Data Visualization	*Tallis Tester, CareVis, GOT*
Plan Rationale / History	*AsbruFlow, Tallis Tester, GLARE, CareVis, GOT*

4.1. Tasks During Design Phase

Plan Generation does not really ask for visualization support in its pure sense, because it tries to create a path of activities in an automatic, semi-automatic, or manual way. However, the process of plan generation and the output of this task can strongly be eased by visual methods, because it is quite difficult to communicate the different steps and the output to domain experts. The same holds for *Plan Verification and Validation*. *Advanced Plan Editing* and *Domain-Specific Annotations* can be supported in two ways: firstly, document-centric approaches (compare Section 1.2) can be used to add structured text and secondly, visual methods can be applied to author and communicate the added or needed parts and to illustrate changes. The other tasks can be supported by selected visualization methods as shown in Table 2.

4.2. Tasks During Execution Phase

Only a few visualization methods are available, which support both the *Plan and Data Visualization* during execution phase (compare the few examples in Section 3). As mentioned earlier in Section , *Plan Modification / Alternatives* is an outstanding task, because it handles maintaining of clinical guidelines and changes in the environment. Therefore, on the one hand, visualization methods are needed, which support adapting the guidelines. On the other hand, visualization methods should ease the monitoring of the patients' health course, the medical environment, and changes thereof in connection with applicable therapeutic actions. The remaining tasks during execution phase could be eased by appropriate methods for *Plan and Data Visualization*.

In summary, there are fewer visual contributions available to support the tasks during design time than during execution time and the visual methods needed to ease the tasks during execution time ask for more features than the other tasks. However, the different tasks can mutually benefit from visual methods of the others.

5. Research Agenda

In this book chapter we presented methods to visualize (1) clinical guidelines seen as plans or activities, (2) patients' data seen as multidimensional information space, and (3) patients' data in connection with clinical guidelines. Contributions of the first two categories are manifold. However, visualizing patients' data in connection with clinical guidelines is a challenging task and only a few approaches are currently available. Many of the visualization methods were developed within the framework of a particular guideline representation language. Therefore, the available visualization methods are mainly oriented towards the specific functionality of the guideline representation language (compare AsbruView, GLARE, and the Tallis Toolset).

According to our findings and various discussions with colleagues coming from different research fields and industry, we can formulate the following research directions:

Visualising the various dimensions of guideline-based care management. We presented different methods to visualize clinical guidelines, patients' data, and the connection thereof (as mentioned above). As we have shown, various approaches exist to visualize patients' data. However, more research needs to be done in the other two categories. On the one hand, visualizing clinical guidelines seen as plans or activities is too much oriented towards the particular features of the guideline representation languages, which asks for more research in the directions of the particular tasks in plan management. On the other hand, visualizing patients' data in connection with clinical guidelines is still an open, but challenging issue.

Designing a science or model of interactions. One very important element of the visual exploration process is interaction(s). Different contributions on various levels exist in the visualization and human-computer interaction community. However, there is no well-accepted science or model of interactions. The guideline-based community could contribute to develop such models for their particular tasks.

Supporting different users, tasks, and data. Visualizations should to be designed and developed according to the different users, their tasks, as well as data and information available. However, the presented visualization methods do not differentiate between these dimensions. Therefore, research should consider these aspects. For example, guideline developers need other visualizations than physicians who are debugging guidelines or real end-users of guidelines.

In-depth evaluation of the visualization methods. As also observed in other fields, the assessment of the usability and utility of the designed visualization methods is partly a neglected issue. However, to illustrate the benefits, a more in-depth evaluation of the visualization methods with particular consideration of the different tasks in plan management is needed.

Acknowledgements

This work is supported by "Fonds zur Förderung der wissenschaftlichen Forschung FWF" (Austrian Science Fund), grant L290-N04.

References

[1] Wolfgang Aigner. Guideline Overview Tool (GOT). Technical Report Asgaard-TR-2001-4, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Austria, 2001.

[2] Wolfgang Aigner and Silvia Miksch. CareVis: Integrated Visualization of Computerized Protocols and Temporal Patient Data. *Artificial Intelligence in Medicine (AIIM)*, 37(3):203–218, May 2006.

[3] Ragnar Bade, Stefan Schlechtweg, and Silvia Miksch. Connecting Time-oriented Data and Information to a Coherent Interactive Visualization. In *Proc. of the 2004 Conf. on Human Factors in Computing Systems (CHI04)*, pages 105–112. ACM Press, 2004.

[4] Stuart Card, Jock Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers, San Francisco, 1998.

[5] Luca Chittaro. Information Visualization and its Application to Medicine. *Artificial Intelligence in Medicine (AIIM)*, 22(2):81–88, 2001.

[6] Luca Chittaro, Carlo Combi, and G. Trapasso. Visual Data Mining of Clinical Databases: An Application to the Hemodialytic Treatment based on 3D Interactive Bar Charts. In *Proceedings of VDM 2002: 2nd International Workshop on Visual Data Mining*, 2002.

[7] Paul A. de Clercq, Johannes A. Blom, Hendrikus H. M. Korsten, and Arie Hasman. Approaches for Creating Computer-interpretable Guidelines that Facilitate Decision Support. *Artificial Intelligence in Medicine (AIIM)*, 31(1):1–27, May 2004.

[8] John H. Gennari, Mark A. Musen, Ray W. Fergerson, William E. Grosso, Monica Crubézy, Henrik Eriksson, Natalya F. Noy, and Samson W. Tu. The Evolution of Protégé: An Environment for Knowledge-based Systems Development. *International Journal of Human Computer Studies*, 58(1):89–123, 2003.

[9] David C. Hadorn. Use of Algorithms in Clinical Practice Guideline Development: Methodology Perspectives. *Clinical Practice Guideline Development: Methodology Perspectives*, 9(95):93–104, Jan. 1995.

[10] Klaus Hinum, Silvia Miksch, Wolfgang Aigner, Susanne Ohmann, Christian Popow, Margit Pohl, and Markus Rester. Gravi++: Interactive Information Visualization of Highly Structured Temporal Data. *Journal of Universal Computer Science, Special Issue on Visual Data Mining*, 11:1792–1805, 2005.

[11] Werner Horn, Christian Popow, and Lukas Unterasinger. Metaphor Graphics to Visualize ICU Data over Time. In *Proceedings of Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-98), Workshop Notes of the ECAI-98 Workshop*, August 25 1998.

[12] Robert Kosara and Silvia Miksch. Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, 22(2):111–131, May 2001.

[13] Tze Yun Leong, Katharina Kaiser, and Silvia Miksch. Free and Open Source Enabling Technologies for Patient-Centric, Guideline-Based Clinical Decision Support: A Survey. *IMIA Yearbook of Medical Informatics 2007, Methods Inf Med*, 46(1):74–86, 2007.

[14] Silvia Miksch. Plan Management in the Medical Domain. *AI Communications*, 12(4):209–235, 1999.

[15] Silvia Miksch, Werner Horn, Christian Popow, and Franz Paky. Utilizing Temporal Data Abstraction for Data Validation and Therapy Planning for Artificially Ventilated Newborn Infants. *AI in Medicine*, 8(6):543–576, 1996.

[16] Mor Peleg, Samson W. Tu, Jonathan Bury, Paolo Ciccarese, John Fox, Robert A. Greenes, Richard Hall, Peter D. Johnson, Neill Jones, Anand Kumar, Silvia Miksch, Silvana Quaglini, Andreas Seyfang, Edward H. Shortliffe, and Mario Stefanelli. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *Journal of the American Medical Informatics Association (JAMIA)*, 10(1):52–68, Jan-Feb 2003.

[17] Catherine Plaisant, Richard Mushlin, Aaron Snyder, Jia Li, Dan Heller, and Ben Shneiderman. LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. In C. G. Chute, editor, *Proceedings of the 1998 American Medical Informatic Association Annual Fall Symposium*, pages 76–80. AMIA, Bethesda, MD, November9–11 1998.

[18] Martha E. Pollack and John F. Horty. There's More to Life than Making Plans: Plan Management in Dynamic, Multi-agent Environements. *AI Magazine*, 20(4):71–84, 1999.

[19] Kristi-Anne Polvani, Abha Agrawal, Bryant Karras, Aniruddha Deshpande, and Richard Shiffman. GEM Cutter Manual. Technical report, Yale Center for Medical Informatics, New Haven, CT, 2000.

[20] Seth M. Powsner and Edward R. Tufte. Graphical Summary of Patient Status. *The Lancet*, 344:386–389, August 6, 1994.

[21] Marek Ruzicka and Vojtech Svátek. Mark-up Based Analysis of Narrative Guidelines with the Stepper Tool. In Katharina Kaiser, Silvia Miksch, and Samson W. Tu, editors, *Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004)*, volume 101 of *Studies in Health Technology and Informatics*, pages 132–136, Prague, Czech Republic, April 2004. IOS Press.

[22] Doug Schaffer, Zhengping Zuo, Saul Greenberg, Lyn Bartram, John Dill, Shelli Dubs, and Mark Roseman. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. *ACM Transactions on Computer-Human Interaction*, 3(2):162–188, 1996.

[23] Yuval Shahar, Dina Goren-Bar, David Boaz, and Gil Tahan. Distributed, Intelligent, Interactive Visualization and Exploration of Time-Oriented Clinical Data and Their Abstractions. *Artificial Intelligence in Medicine*, 38:115–135, October 2006.

[24] Yuval Shahar, Silvia Miksch, and Peter Johnson. The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. *Artificial Intelligence in Medicine*, 14:29–51, Sept. 1998.

[25] Yuval Shahar, Ohad Young, Erez Shalom, Alon Mayaffit, Robert Moskovitch, Alon Hessing, and Maya Galperin. DEGEL: A Hybrid, Multiple-Ontology Framework for Specification and Retrieval of Clinical Guidelines. In Michel Dojat, Elpida Keravnou, and Pedro Barahona, editors, *Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe (AIME 2003)*, volume 2780 of *LNAI*, pages 122–131, Protaras, Cyprus, 2003. Springer Verlag.

[26] Richard N. Shiffman, Bryant T. Karras, Abha Agrawal, Roland Chen, Luis Marenco, and Sujai Nath. GEM: A Proposal for a More Comprehensive Guideline Document Model Using XML. *Journal of the American Medical Informatics Association (JAMIA)*, 7(5):488–498, 2000.

[27] Society for Medical Decision Making. Proposal for Clinical Algorithm Standards. *Medical Decision Making*, 12(02):149–154, April-June 1992.

[28] Rory Steele and John Fox. Tallis PROforma Primer – Introduction to PROforma Language and Software with Worked Examples. Technical report, Advanced Computation Laboratory, Cancer Research, London, UK, 2002.

[29] David R. Sutton and John Fox. The Syntax and Semantics of the PROforma Guideline Modeling Language. *Journal of the American Medical Informatics Association (JAMIA)*, 10(5):433–443, 2003.

[30] Paolo Terenziani, Stefiana Montani, A. Bottrighi, M. Torchio, G. Molino, and G. Correndo. The GLARE Approach to Clinical Guidelines: Main Features. *Stud Health Technol Inform*, 101:162–166, 2004.

[31] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press, 2005.

[32] Eduard R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.

[33] Peter Votruba, Silvia Miksch, and Robert Kosara. Facilitating Knowledge Maintenance of Clinical Guidelines and Protocols. pages 57–61. AMIA, IOS Press, 2004.

[34] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, San Francisco, CA, 2004.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008
© 2008 The authors and IOS Press. All rights reserved.
doi:10.3233/978-1-58603-873-1-160

Chapter 9
Compliance with Clinical Practice Guidelines

Silvana QUAGLINI[1]
Department of Computer and System Science, University of Pavia, Italy

Abstract. Compliance with clinical practice guidelines is a challenging topic because it depends on a variety of factors, some related to guidelines themselves, some related to users, and some to the implementation context. Among the former are guideline quality, purpose and implementation modality. Among the user-related factors are attitude to behavioural changes, authority interventions to foster adherence and eventually the type of users (general practitioners, hospital professionals, home caregivers, patients, etc.). Context is also crucial because organisational issues, such as lack of resources, can hamper guideline implementation and sometimes the original guideline intention is overridden by the guideline adaptation to a certain setting. This chapter analyses these factors and discusses their implications for the development of computerised decision support systems. Moreover, it gives examples of non-compliance detection and analysis in a specific real-world computerised guideline implementation, facing both methodological and practical issues.

Keywords: compliance, clinical practice guidelines, health outcomes, economical outcomes

Introduction

This chapter discusses the behaviour of guidelines' users with respect to guidelines suggestions and describes how this behaviour can be captured by a computerised decision support system and exploited to improve guidelines' implementation. However, the chapter will first describe guideline-related issues independently from computerisation, because understanding the different human behaviours that may be observed in clinical routine is of paramount importance for developing computer-based decision support systems with high chance of success, and for continuously improving already developed systems.

User behaviour must also be analysed in the light of the user *type*: the most common users of guidelines are physicians or other healthcare professionals, and the main focus, in this chapter, will be on them. However, for the sake of completeness, we must mention that patients and caregivers are often involved, and scientific societies may deliver different versions of the same guideline for different users (see for example the AHRQ[2] repository, where some guidelines come with a version for

[1] Corresponding Author: Silvana Quaglini, Department of Computer and System science, University of Pavia, via Ferrata, 1, 27100 Pavia, Italy; E-mail: silvana.quaglini@unipv.it

[2] Agency for Healthcare Research and Quality, www.ahrq.gov

patients/caregivers). This aspect is crucial when developing a computer-based version of a guideline. In fact, the overall system must be able to carry out the so-called "continuity of care", where patients and their caregivers become *responsible* subjects with respect to their own health and must be fostered and helped to comply with the prescribed home treatment. Patients' compliance is behind the scope of this Chapter, and the reader is referred to the extensive literature, in particular to studies concerning elderly patients, for whom compliance is also related to safety and for this reason it is particularly important [1,2,3].

Let us now introduce some terminology. Given a recommendation from a guideline, a user may act according or not, and in both cases he may agree (in principle) or not with the recommendation itself. As a matter of fact, a user who disagrees with a guideline could be "forced" to use it, while a user who in principle agrees with a guideline could find some problems that prevent its recommendations being achieved. Throughout this chapter, the terms "compliance" and "adherence" will mean "acting according to the recommendations of a guideline", while "non-compliance" will be used to mean any deviation from its recommendations, for whatever reason, and independent of the modality of the implementation of the guideline . In fact, when a user acts differently from a guideline recommendation, motivations can be found in his disagreement with the guideline, or in other types of barriers, and these motivations will be defined and discussed in the following sections. Further discussion on terminology in this area is reported in [4].

The chapter is organised as following: Section 1 illustrates actions taken by the medical and medical informatics communities, in recent years, to foster compliance; Section 2 illustrates some compliance issues through case studies; Section 3 analyses motivations for non-compliance within a more formal framework; Section 4 shows how to rate non-compliance according to their severity; Section 5 illustrates how information technology can improve compliance and allow useful non-compliance analysis; finally, Section 6 shows how to exploit computerised data collection to analyse correlation between compliance and outcomes. Research Agenda will conclude the chapter. Therefore, we do not discuss the non-compliance that may arise when guidelines are made site-specific, i.e. when they are adapted to the actual clinical settings. Adaptation, which is almost always necessary, could lead to non-compliance with the original guideline intention. Throughout the Chapter, we assume that guidelines have been already adapted by official committees of the specific country, or region, or healthcare agency, down to those responsible for a specific hospital. For further consideration of the topic of adaptation, see Chapter 7 and additional literature [5,6,7,8,9,10].

1. Background

Searching PubMed[3] using "clinical practice guidelines" as keyword, the earliest indexed papers date from the beginning of the 1970's, but the number of papers in the field starts growing dramatically in the late 1980's. By adding "efficacy", "compliance", "effectiveness", "adherence", etc. as keywords, earliest papers date from the beginning of the 1990's. This suggests that after about a decade from large-scale guidelines' development, issues arose with their real-world implementation, penetration into

[3] PubMed, U.S. National Library of Medicine, www.pubmed.org

clinical routine, and impact on health and economic outcomes. The title of [11] ("...guidelines: promise or illusion?") is self-explanatory. Both researchers and authorities, who put great confidence in the potential of guidelines, realised that compliance with guidelines was (and continues to be) poor [12,13,14,15, 16, 17].

To improve compliance, in some countries, attempts have been made to introduce strictly mandatory guidelines: for example, from 1993 onwards, in France this policy was tried for general practitioners, with a system of fines for doctors who did not comply [18]. These guidelines aimed to limit the prescription of redundant and costly drugs, tests, and procedures by fining, up to about 1,715 Euros, physicians who overprescribed. Despite the official implementation of this policy, awareness and knowledge of the regulatory guidelines among French family physicians has been weak. Durieux et al. [19] explain this finding by the difficulties in controlling physicians in the use of a large number of guidelines, and express concerns about the efficacy of this policy in the long term.

There is now a common agreement that mandatory guidelines and financial disincentives (or incentives) can be used to change physician behaviour only if they are simple, well explained, and do not raise any ethical or professional conflict. Moreover, they must be always accompanied by educational intervention and, if necessary, organisational changes.

Starting about 20 years ago, issues on the medico-legal implications of guidelines, an aspect that obviously affects compliance, were also tackled [20]. A United States court established rules for both guideline developers and users, ruling that developers can be held liable for faulty guidelines and physicians cannot indiscriminately avail themselves of their "personal clinical judgement" to justify their non-compliance with common guidelines. On the other hand, judges are well-aware that adherence to guidelines does not automatically imply "perfect practice"; thus the current common sense is that controversies must be solved with prudence, critically evaluating every specific case in the light of the guideline quality, flexibility, and scope of application.

Among physicians there is no common opinion about the benefits and drawbacks of guidelines from the legal point of view: for example, in the U.K., a survey carried out by the National Health and Medical Research Council of 150 surgeons to determine their views on the clinical practice guidelines for breast cancer, showed that 37% felt that the guidelines would increase their exposure to medico-legal problems, while 41% felt that they would protect them from those problems [21]. In general, it is agreed that documenting clinical activities, which is always a good practice, becomes of paramount importance when a clinician practises outside the guidelines [22], and this argument points out the necessity of using computerised support in routine clinical practice, since it facilitates such documentation (see Section 5).

Another aspect of guideline compliance is the perceived usefulness of the guideline itself. The recent SIGN[4] methodology for developing new guidelines [23] consists in considering only proposals derived from collaborative networks of experts and relevant stakeholders, coordinated by an Advisory Group representative of the U.K. National Health Service. Standard application forms must be used to make the proposal, and multidisciplinary groups evaluate whether there is a general agreement about the need for guidelines in that field. In such a way, it will be much more likely that the final users will comply with the guidelines produced because, in some sense, a previous consensus has been achieved.

[4] Scottish Intercollegiate Guideline Network (www.sign.ac.uk)

Finally, we consider compliance related to the modality of guidelines' delivery. There are several studies indicating that computerised decision support systems, with respect to paper-based versions or hypertexts, can improve clinicians' compliance [24,25,26,27]. But we must remember the lesson learned from medical expert systems: despite the great emphasis that the scientific Artificial Intelligence community put into them in the 1970's, most of them failed in their objective to enter medical practice. In [28], the authors describe 30-years of experience with a variety of expert systems and concluded that "we should stop trying to make a computer act like a diagnostician and concentrate instead on ways of making computer-generated relevant information available to physicians as they make decisions". These failures were not due to poor reliability and accuracy of the recommendations produced, but mainly to the lack of attention given to the systems' integration into clinical practice. Human-computer interaction (in a broad sense) is now widely recognised as the key factor for making decision support systems acceptable to healthcare professionals [29]. Sim et al [30] want to see "workflow–sensitive implementations of clinical decision support systems", meaning that they must provide bedside assistance and be fully integrated with the work processes of clinicians. Of course, this in turn means that computers should be pervasive in the work setting, and this is not yet a reality for the most healthcare organisations. However, in highly-computerised settings, good acceptance and improved compliance have been shown [25, 31]. An example will be provided in Section 5.

2. Case Studies

In this section we consider compliance as part of the real-world implementation of a guideline. In the first case study, we want to stress that deep knowledge of the medical area addressed by the guideline is necessary in order to analyse compliance and understand motivations for non-compliance. Since guidelines are not "cookbook medicine", various approaches are needed to study compliance with different guidelines, and many factors must be considered, such as the implementation time, location, the skill of the personnel and healthcare organisation. Guideline computerisation must consider all these factors.

Gilligan et al. [32] describe an American study on the compliance with guidelines for breast cancer treatment. A cohort of about 1,045 surgeons that treated 9,449 women older than 64 years was considered. The aim of the study was not only to verify surgeon compliance, but also to investigate whether it was correlated to some surgeon's characteristics. The guidelines, published in 1990 by the National Institute of Health [33], basically suggested, in the case of conservative surgery, axillary lymph node dissection and radiotherapy or, in case of radical surgery, only axillary lymph node dissection. It must be considered that case enrolment finished in 1996, and some of the current techniques, such as the examination of the sentinel lymph node, had not yet entered clinical practice, nor were they suggested by any official guideline.

For classifying surgeons, the following factors were considered: age, sex, number of surgical interventions performed, having or not a specialty, belonging or not to University staff. The result of the study was that about 75% of patients received the guideline suggested treatment. However, a great difference was observed between the two types of surgery: in case of mastectomy, guideline compliance was about 92%, while in case of conservative surgery it was only 55%. In fact, lymph node dissection was often not performed, particularly when the surgeon was a female or a University professor. Guideline compliance was directly correlated with the number of interventions performed. The age of the surgeon was not a significant factor.

How should we interpret these non-compliances? Of course *non-compliance* does not necessarily mean *bad treatment*. We mentioned that in order to understand non-compliance, medical knowledge in the specific area is mandatory. In this case study, for example, since most tumours were in the initial phase, it is probable that several surgeons, maybe those University staff members, were already knowledgeable about the sentinel lymph node examination, having privileged access to scientific results not already published in official guidelines (e.g., through participation in research collaboration groups, conferences, etc.). Or they could have decided to skip the dissection because they prescribed an adjuvant therapy. Also, the positive correlation between a surgeon's experience and guideline compliance deserves comment: breast cancer surgery is not difficult, thus the advantage for a patient to be treated by an expert surgeon probably lies in the fact that he/she belongs to a medical centre performing a high number of interventions. These centres are better organised, with an established routine for both hospital stay and follow-up. It is worth stressing the important role of a good organisation in fostering and facilitating compliance with official guidelines.

Additional case studies can be reported, each one pointing to specific issues:

- health care system inefficiency, as described in [34], where the authors analyse the data of patients admitted with chest pain who were not discharged according to a practice guideline. Inefficiency was due to waiting for tests, procedures, consultation, or results, and also to social issues (e.g., waiting for nursing home placement);
- demanding a change in existing practice routines. In a study on general practitioners' compliance, Grol [35] showed that recommendations which demanded a change of behaviour were followed in 44% of decisions and those that did not in 67% of decisions;
- lack of effective leadership, as described in [36], where institutional decisions to ignore guidelines issued by Centers for Disease Control and Prevention are claimed to have caused serious spread of infection;
- particular, or critical, conditions poorly addressed in the guidelines, as described in [37], where "... Womens' age appears to be a major explanatory variable predicting lack of physician's compliance with consensual norms".

The next Section addresses the problem of reasoning about non-compliance within a more theoretical and general framework.

3. Reasons for Non-compliance

From Section 1 it seems that several interventions to improve guideline compliance have been established without investigating the true motivation for non-compliance. Indeed, this should be the first step to improve both compliance and guidelines. In a review of 2002, Farquhar et al. [38] summarise results from a total of 11,611 responses to various surveys. Table 1 shows that the majority of respondents share a good opinion of guidelines, but several and important drawbacks are also perceived by a considerable percentage of physicians. In fact, despite the large investment in development and dissemination, we have already mentioned that physicians' adherence to guidelines has not been as high as expected. There are several reasons for this, and the most frequent ones can be argued from Table 1 (items 4-7). The next two subsections will analyse more deeply two of the more frequent motivations for non-compliance, namely user disagreement with the recommendations and the unclear purpose of the guideline.

Table 1 - Physicians opinion about seven propositions on Guidelines (adapted from [38])

1. helpful source of advice	75%	5. reduced physician autonomy, over-simplification or "cookbook" medicine	34%
2. good educational tools	71%	6. will increase litigation or disciplinary action	41%
3. intended to improve quality of care	70%	7. were intended to cut healthcare costs	53%
4. too rigid to apply to individual patients	30%		

3.1. Disagreement with Guideline Recommendations

The simplest motivation for non-compliance is disagreement with the guideline itself. One of the arguments of guidelines detractors is related to the source of scientific evidence. Clinical trials are claimed to be the best methodology to provide evidence for recommendations to include in a guideline. On the other hand, it is well-known that the conditions in which clinical trials are carried out are different from the situation in the real-world: within a trial, patients undergo a tight follow-up; trials are funded, so there are no problems in acquiring drugs, instrumentation, and for patient's out-of-pocket costs, etc.; finally, it is known that the behaviour of professionals and patients involved in a scientific study is positively biased ("Hawthorne" effect, see [39,40]): if an individual participates in a research study, he may be motivated to perform better. Thus physicians may not fully trust guidelines, because of the uncertainty about the true efficacy of the trial results once they are transferred to clinical routine [41], and they could prefer to continue with their practice and "resist" the proposed change.

However for physicians who are not "guidelines detractors", there may be reasons for disagreement. Guidelines recommendations represent the "optimal" practice for the "average" patient, but no guideline can delineate all of the clinical subtleties of a patient. It is clear that intra- and inter-patients variability justifies medical behaviours different from the recommended one, since guidelines rarely are so complete as to capture all the possible scenarios that can arise during a patient's clinical pathway.

Another type of disagreement occurs when a physician trusts a guideline recommendation, and applies it, but to a patient that does not belong to the guideline target population. The reason is still related to clinical trials: to be rigorous, a trial result can only be transferred to patients similar to those enrolled i the trial itself. For example, the recommendation about the thrombolitic drug r-TpA, in the SPREAD[5] guideline for stroke management [42], is limited to people less than 80 years old, because no trials yet exist for older people. The SPREAD guideline explicitly claims that being older than 80 years is a contraindication for the drug. Thus, administering it to those patients is formally considered to be a non-compliance. Of course, a physician could decide to do it for an 81 year old patient, if he estimates a lower "biological age"

[5] www.spread.it, this guideline will be mentioned again in this chapter, as a training exemplar.

for that patient, without in fact deviating from good clinical practice and from the intention of the guideline.

3.2. Non-compliance Due to Unclear or Ambiguous Guideline Purpose

In order to propose, implement, and then evaluate a guideline, its purpose must be clear to the final users. For example, in the classical definitions of what guidelines are, from that given by the Institute of Medicine in 1990 ("Systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances"), to more recent ones, *economical costs* are not mentioned. In fact, usually, cost of care is not the key motivation for the development of a guideline. Nevertheless, the inclusion of economic appraisals in the guideline implementation process is more likely to allow clinicians and consumers to make cost-effective choices on health care options [43]: costs decrease if guidelines eliminate or limit the number of unnecessary expensive activities which are often performed in clinical routine without any scientific basis. But, given the current well-known scarcity of resources, some physicians could argue that governmental healthcare organisations promote guidelines with the main objective of containing costs, with poor attention to the quality of the delivered care, and this gives them a justification for non-compliance [44,45]. The relationship between guideline introduction and cost containment is well synthesised in the outstanding paper of Greco and Eisenberg [46]: "What causes physicians to change the way they practice? This question is especially important today because physicians' decisions influence not only the health of their patients but also the cost of their care. Thus, the ability to change physicians' practices could improve the quality of health care while controlling expenditures". As a matter of fact, physicians are more and more urged to pay attention to economical aspects, and often feel that their medical activity is limited by this obligation.

On the other hand, even healthcare administrators can be uncertain whether to promote or not a guideline implementation, mainly when it incurs new costs. They can be interested in knowing whether this intervention affects the total cost of care: implementing new procedures could involve acquiring instrumentation, performing additional tests or visits, or employing more personnel. It is very important to provide decision makers with estimates of the incremental cost-effectiveness ratio of the guideline[6]. Without such estimates, having a suspicion of increased costs could discourage them from supporting the implementation of a guideline. As an example, the economic evaluation of the implementation of the SPREAD guideline [47] showed that complying with the guideline is cost-saving. Similarly, in the past, Clayton [48] showed that adherence to guidelines and protocols may reduce health-care costs by up to 25%. In fact, several non-compliances consist of omissions, bad choice of drugs, and violation of time constraints, that in turn cause adverse health outcomes that prolong the hospital stay and treatments, or require re-hospitalisations, thus increasing medium- and long-term costs. A pilot implementation study, showing a favourable cost/effectiveness ratio, is useful in convincing both administrators and physicians of the good impact of a guideline in a real-world setting [35].

Going back to the purpose of a guideline, whatever it is, it should be worth the guideline authors declaring it. In addition to the previously mentioned advantages, a clear purpose facilitates the design of validation studies. For example, if a guideline is

[6] Incremental cost-effectiveness analysis measures the cost per unit of health outcome gained

intended to reduce mortality, it should be declared if mortality is to be measured in the short or longer term, in such a way that a correct survival study could be carried out, and the necessary effort foreseen from the beginning of the implementation.

We must stress at this point that guideline implementation with computer support is central in performing reliable analyses (see Sections 5 and 6). To formally store a guideline intention, authors can comply, for example, with the Guideline Elements Model (GEM), an XML-based guideline document model that stores and organizes the heterogeneous information contained in guidelines [49]. As an example, Figure 1 is drawn from the Guide Project [8], that includes an interface to store guideline information according to the GEM criteria. In particular "Category", "Rationale", "Objective" and "Potential risks" are the attributes used to test the effectiveness of the guideline implementation. The Guide Project adopts the guideline categories of the National Guideline Clearinghouse[7], and objectives can be split into health-related and cost-related outcomes.

Other languages for the representation of guideline intentions are provided by the SAGE project [50], and the ASBRU project [51].

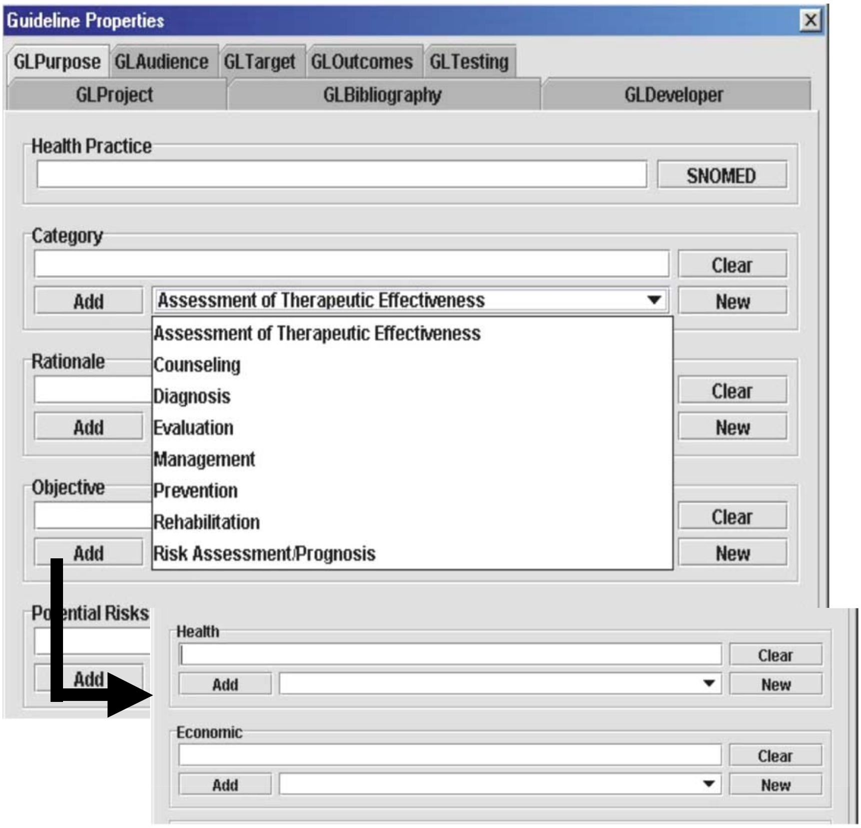


Figure 1 - Storing the general characteristics of a guideline according to the GEM model. Objectives may be related either to the health outcome or the economic one.

[7] www.guideline.gov

3.3. Other Motivations for Non-compliance

In addition to disagreement, poor confidence, and poor perceived guideline usefulness, more objective conditions could hamper its implementation. Since a guideline mainly suggest *actions*, building a semantic network, with *action* as the core, may help in reasoning on non-compliance. One such network is shown in Figure 2.

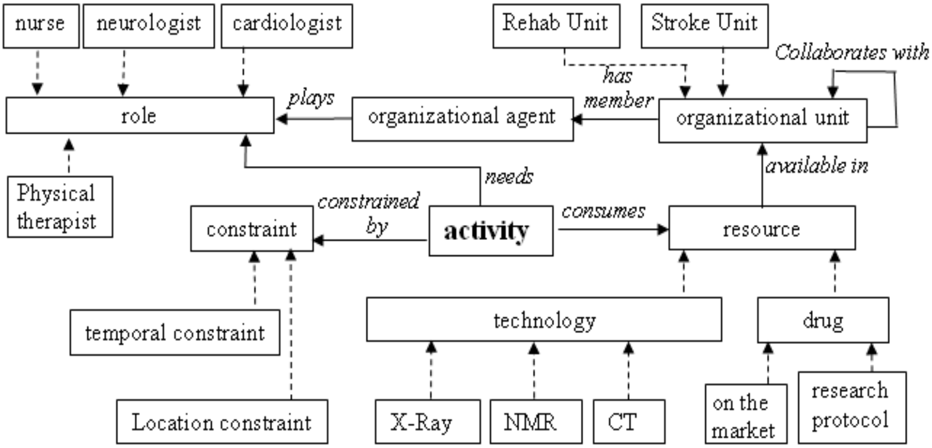


Figure 2 - Semantic network relating the "activity" to the environmental conditions necessary for its execution. Dashed arcs represent *is-a* relationships.

It has been drawn up in order to analyse physicians' behaviour with respect to the SPREAD guidelines; thus some leaves are typical of actions for stroke patients. Coupling this network with the organizational model of the clinical unit where compliance has to be tested, allows guessing possible causes of non-compliance (an organisational model describes a unit in terms of resources and responsibilities). For example, if a guideline recommends an NMR-scan and the Unit does not hold that instrumentation, nor does it collaborate with other Units holding that instrumentation, the motivation for non-compliance can be inferred automatically.

3.4. Classification of the Motivations for Non-compliance

Analysis of non-compliance is very useful in improving both users' behaviour and further guideline versions.

Since, as discussed in the previous sections, a non-compliance always comes with a motivation, it is sensible to define a motivation taxonomy. The classification in Table 2, which partially comes from the semantic network in Figure 2, is conceived in such a way that non-compliance collection and analysis produces feedback for well-defined responsible persons. Motivations have been classified into four main categories, capturing most problems that can arise during clinical routine. For sake of simplicity, Table 2 represents only the first sublevel of the four main categories, but the whole taxonomy is more detailed [52]. Once non-compliances have been labelled according to this taxonomy, reports can be produced and sent to responsible persons or units, who may be able to solve or manage the problem. Of course, the more specific the level, the more appropriate can be the feedback.

Table 2- Classification of the motivations for non-compliance

1-Organisational Problems	1.1 Lack of personnel
	1.2 Lack of non-human resources (Instrumentation / Drugs)
	1.3 Data/Information flow problems, excluding software problems
2-Technical Problems	2.1 Instrumentation fault
	2.2 Software bugs/malfunctioning (causing errors in data interpretation)
3-User-related issues	3.1 Disagreement with the guideline
	3.2 Participation in a Research Protocol (justifying deviation from guidelines for research purposes)
	3.3 Lack of adequate skill
	3.4 Medical error
4-Patient-related issues	4.1 Lack of consensus
	4.2 Unpredictable patient's finding making the guideline no longer appropriate
	4.3 Early patient discharge (including death)

For example, a the recurrent unavailability of a particular instrument will be notified to the clinical engineering department; frequently missed data in the electronic clinical chart will be notified to the EDP department, leading to a better data communication infrastructure; more interestingly, frequent disagreement on a certain recommendation may be communicated to the board that is maintaining the guideline (i.e. an expert panel of a scientific society), for the further guideline revisions. It is likely that the recommendation is not suitable for the specific site, or that new evidence has arisen in the meantime, making the recommendation obsolete. In Section 5 we show how this taxonomy is exploited within a decision-support system.

4. Rating the Severity of Non-compliances

To rate non-compliances according to their severity, we take two points of view: the first is related to the recommendation itself, the second to the consequence or outcome of the non-compliance. Whichever rating scheme is adopted, it is important that computerised guidelines take it into consideration: on the one hand, modulating the non-compliance advice according to the non-compliance severity improves the system acceptance; on the other, statistics about non-compliance will be more sophisticated and sensible.

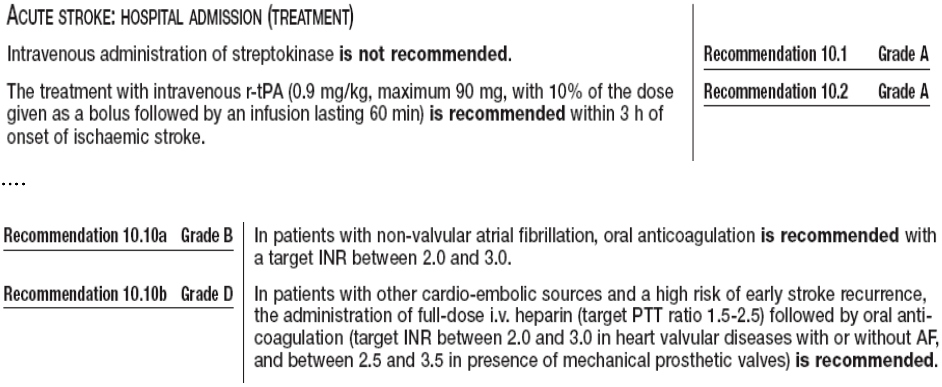
4.1. Rating Non-compliances According to the Scientific Degree of the Recommendation

Good quality guidelines rate their recommendations, according to the type and soundness of the research studies supporting them: Chapter 1 reports on grading methodologies. Very briefly, we recall here that levels of scientific evidence have been proposed since 1979 and have evolved over the ensuing years [53,54,55,56,57]. In particular, during recent years, and accordingly to principles of the SIGN (Scottish Intercollegiate Guideline Network) [58], methodologies have been developed to integrate scientific evidence levels with indicators of direct applicability of the results of a study to the target population. The resulting rating is called "Grade of recommendation" (see Table 1 and again Chapter 1), that include chance of effective implementation.

Table 3- Grade of recommendation delivered by SIGN

A	At least one meta analysis, systematic review, or RCT rated as 1++, and directly applicable to the target population; *or* A systematic review of RCTs or a body of evidence consisting principally of studies rated as 1+, directly applicable to the target population, and demonstrating overall consistency of results
B	A body of evidence including studies rated as 2++, directly applicable to the target population, and demonstrating overall consistency of results; *or* Extrapolated evidence from studies rated as 1++ or 1+
C	A body of evidence including studies rated as 2+, directly applicable to the target population, and demonstrating overall consistency of results; *or* extrapolated evidence from studies rated as 2++
D	Evidence level 3 or 4; *or* extrapolated evidence from studies rated as 2+; or Evidence from studies classified as - (minus), regardless of the level
Good Practice Point	Recommended best practice based on the clinical experience of the guideline development group, without research evidence

When a guideline is clearly annotated with levels of scientific evidence and/or grade of recommendation (see an example in Figure 3, drawn from the SPREAD web site), it is probable that a physician is more motivated to adhere to highly rated recommendations.



ACUTE STROKE: HOSPITAL ADMISSION (TREATMENT)

Intravenous administration of streptokinase **is not recommended.** — Recommendation 10.1 Grade A

The treatment with intravenous r-tPA (0.9 mg/kg, maximum 90 mg, with 10% of the dose given as a bolus followed by an infusion lasting 60 min) **is recommended** within 3 h of onset of ischaemic stroke. — Recommendation 10.2 Grade A

....

Recommendation 10.10a Grade B — In patients with non-valvular atrial fibrillation, oral anticoagulation **is recommended** with a target INR between 2.0 and 3.0.

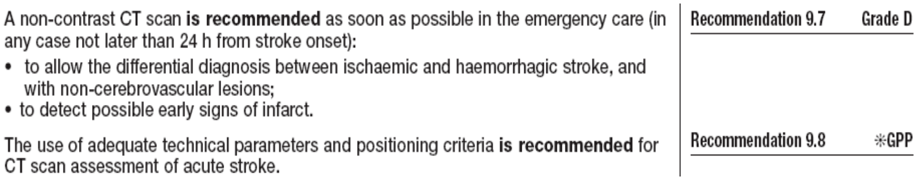
Recommendation 10.10b Grade D — In patients with other cardio-embolic sources and a high risk of early stroke recurrence, the administration of full-dose i.v. heparin (target PTT ratio 1.5-2.5) followed by oral anti-coagulation (target INR between 2.0 and 3.0 in heart valvular diseases with or without AF, and between 2.5 and 3.5 in presence of mechanical prosthetic valves) **is recommended.**

Figure 3 - Two portions of the SPREAD guideline for the management of stroke patients. Every recommendation is reported with its grade of recommendation

4.2. Rating Non-compliances According to their Consequences

Degree of scientific evidence and grade of recommendation are not necessarily related to the importance or relevance of the recommendation. As remarked by SIGN, the grading "relates to the predictive power of the study designs from which that data was obtained. Thus, the grading assigned to a recommendation indicates to users the likelihood that, if that recommendation is implemented, the predicted outcome will be achieved".

For example, a therapeutic recommendation about pain relief could come with the highest degree (e.g. administering the drug is effective in 99% of cases) but, it is not life-threatening to disregard it. On the other hand, a diagnostic recommendation, such as performing CT scan (Computer Tomography) within 24 hours from the onset of stroke symptoms, comes with a lower grade, but missing it may compromise the patient's health outcome (Figure 4). It is probable that physicians often reason according to the utility theory [59], weighing the probability of an event with the severity of its consequences. This must be taken into account when reasoning about non-compliance.



A non-contrast CT scan **is recommended** as soon as possible in the emergency care (in any case not later than 24 h from stroke onset):

- to allow the differential diagnosis between ischaemic and haemorrhagic stroke, and with non-cerebrovascular lesions;
- to detect possible early signs of infarct.

Recommendation 9.7 Grade D

The use of adequate technical parameters and positioning criteria **is recommended** for CT scan assessment of acute stroke.

Recommendation 9.8 ✻GPP

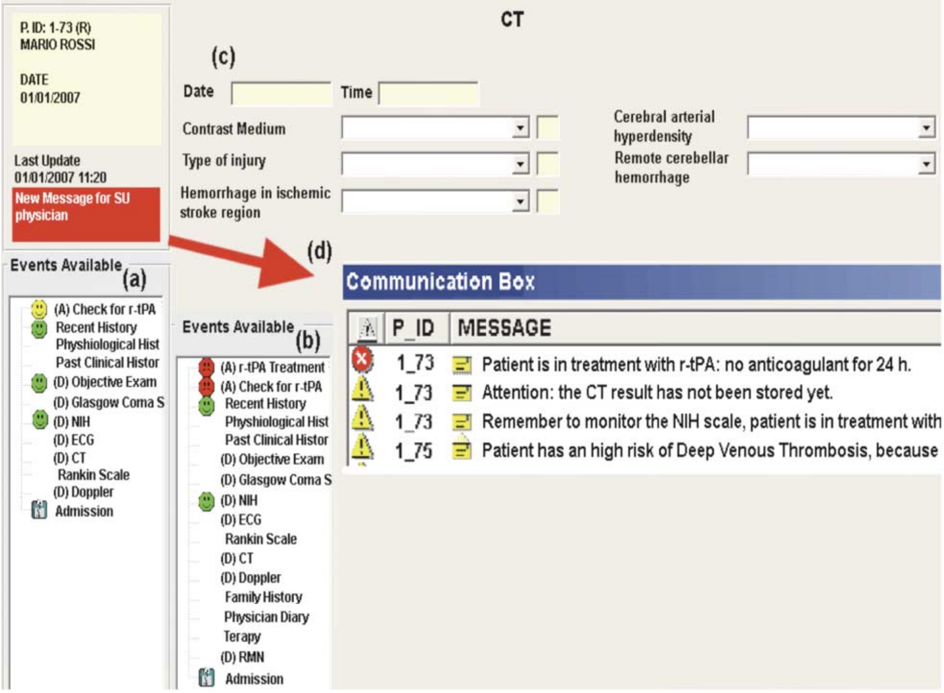
Figure 4 - A SPREAD recommendation with low grade but extremely important to diagnostic purposes

5. Facilitating and Assessing Guideline Compliance: the Role of Information Technology

As mentioned in Section 1, there is now evidence about the improvement in compliance provided by computer-based guideline implementation (decision support systems) with respect to paper-based guideline delivery. Lobach [24] showed that decision support based on a clinical practice guideline is an effective tool for assisting clinicians in the management of diabetic patients. Gandhi [60] showed that primary care physicians perceived the utility of electronic reminders for both routine health maintenance and chronic disease management, and that most of physicians preferred receiving reminders in an electronic format rather than on paper.

Since running a computer-based guideline without relying on a computerised clinical chart (CCC) (for accessing patient's data) is neither very realistic nor efficient, in this paragraph we assume that a CCC is available. This not only allows the guideline inference engine to be provided automatically with data, but also facilitates guideline compliance. Let us consider for example the implementation of the SPREAD guidelines. Here, the guideline has been fully integrated within the CCC interface, through the creation of *smart forms,* such as the form shown in Figure 5. Its characteristics are:

- highlighting critical information. Not all of the data of a CCC are important for guideline implementation. The form indicates, by using a different colour, which data are really necessary. Tasks to be done are also emphasized and shown in order of urgency, which changes from time to time according to the data collected. It is clear that collecting the correct data is the first step towards compliance;
- showing reminders. A window can be open when the "communication button" turns to red, reporting all the current recommendations, for all patients, still not fulfilled.



	P_ID	MESSAGE
☒	1_73	Patient is in treatment with r-tPA: no anticoagulant for 24 h.
⚠	1_73	Attention: the CT result has not been stored yet.
⚠	1_73	Remember to monitor the NIH scale, patient is in treatment with
⚠	1_75	Patient has an high risk of Deep Venous Thrombosis, because

Figure 5 - Fostering compliance: to-do list is patient-tailored and varies with time (a,b); yellow fields indicate data necessary for guideline implementation (c); a communication button turns to red when there are new messages (d).

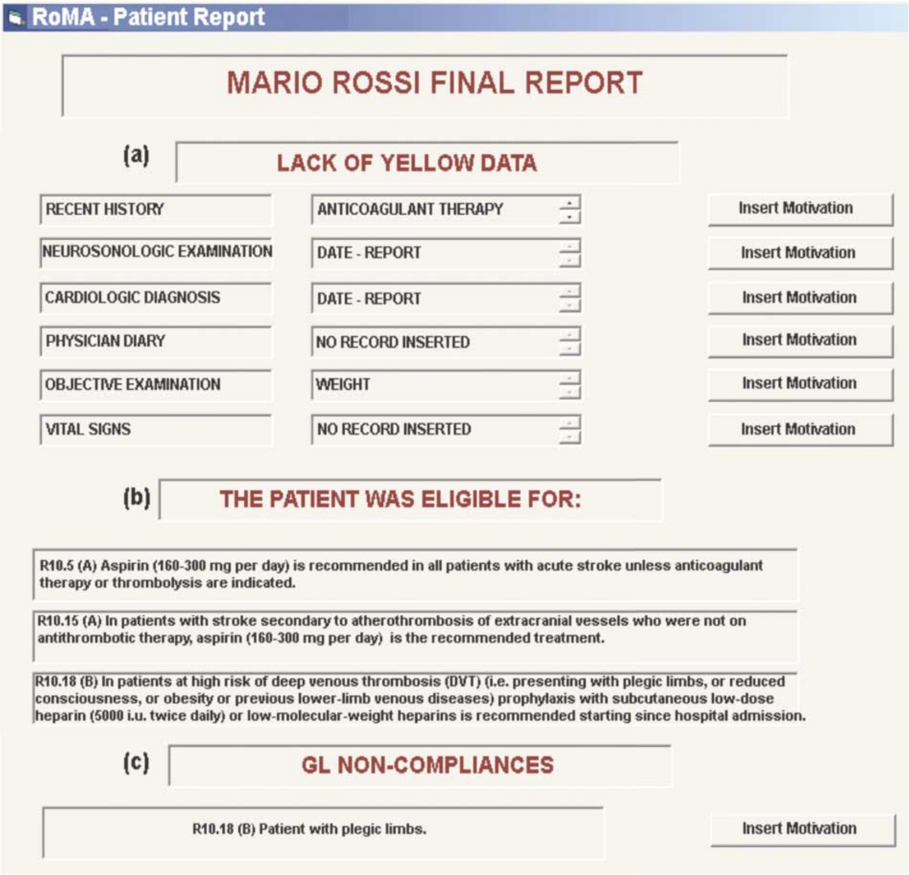
It should be argued that a guideline inference engine integrated within a CCC, would allow detecting non-compliance in real-time and also asking for motivations in real time. However, this is not so straightforward. In a real-world healthcare setting, it is extremely difficult to detect non-compliance in real time. There are too many variables that affect timely data storage (first of all unavailability of ubiquitous data input devices) and, consequently, automatic detection of non-compliance. This is particularly true in acute patient units, where timely actions at patients' bed have the priority on timely computer data input.

Thus, arguing that physicians might justify their compliance in real time has some drawbacks:

- a detected non-compliance could be not real, i.e. an action has been performed but related data have not yet been entered in the CCC: this causes the inference engine to detect an apparent non-compliance;
- the physician perceives the system as not only as a gentle reminder, but also as a controller;
- if he is in a hurry, the physician may have no time to enter the justification;
- as a consequence of the points above, the justification provided may be not reliable.

For these reasons, non-compliance must be managed in a less invasive way. A suitable point of time is the patient discharge. When physicians prepare for a discharge, they are usually in a team and not in a hurry. Moreover, writing down the discharge letter requires summarizing the patient care process, and reasoning about it. Thus, presenting at this point of time the non-compliance list (detected by the system automatically from the CCC data) has two advantages: overcoming the previous problems and facilitating reasoning about the case.

Figure 6 shows the user's interface of RoMA (Reasoning on Medical Actions), a ancillary tool for non-compliance management, coupled with a decision support system for stroke patients [31].



MARIO ROSSI FINAL REPORT

(a) LACK OF YELLOW DATA

RECENT HISTORY	ANTICOAGULANT THERAPY	Insert Motivation
NEUROSONOLOGIC EXAMINATION	DATE - REPORT	Insert Motivation
CARDIOLOGIC DIAGNOSIS	DATE - REPORT	Insert Motivation
PHYSICIAN DIARY	NO RECORD INSERTED	Insert Motivation
OBJECTIVE EXAMINATION	WEIGHT	Insert Motivation
VITAL SIGNS	NO RECORD INSERTED	Insert Motivation

(b) THE PATIENT WAS ELIGIBLE FOR:

R10.5 (A) Aspirin (160-300 mg per day) is recommended in all patients with acute stroke unless anticoagulant therapy or thrombolysis are indicated.

R10.15 (A) In patients with stroke secondary to atherothrombosis of extracranial vessels who were not on antithrombotic therapy, aspirin (160-300 mg per day) is the recommended treatment.

R10.18 (B) In patients at high risk of deep venous thrombosis (DVT) (i.e. presenting with plegic limbs, or reduced consciousness, or obesity or previous lower-limb venous diseases) prophylaxis with subcutaneous low-dose heparin (5000 i.u. twice daily) or low-molecular-weight heparins is recommended starting since hospital admission.

(c) GL NON-COMPLIANCES

R10.18 (B) Patient with plegic limbs.

Figure 6 - The interface for visualising non-compliances and collecting motivations

At the patient discharge, a set of queries is activated on the electronic patient record, and a sheet is filled and printed as illustrated in the picture. The report shows the important missing information (that very probably is reported on the paper-based clinical chart, but not in the computerised one), and the list of non-compliances detected. It gives the physician the possibility to provide motivations that can be useful for further revision of the guideline, as discussed before (note that nothing is mandatory the entry of motivations depends on the user's willingness to do so). In fact, the "insert motivation" button, on the bottom right, opens the taxonomy described in Table 2, and the user can choose an item and comment further on. Moreover, in the same form, there is a link to the SPREAD guideline recommendation text, and the user can compare the original guideline text with the actual formalisation, to check whether the formalised rule is a correct interpretation of the guideline intention. This facility is particularly effective during the debug phase of a computerised guideline. As a matter of fact, when non-compliances are detected automatically, *false* non-compliances could be found, and not only for lack of data, as discussed above. As a matter of fact, guideline computerisation requires translating recommendations from free text to much more formal representations, such as production rules, and this formalisation is difficult, because natural language may contain ambiguities. It is not rare for different physicians to give different interpretations of the same portion of text. Thus, it may happen that a formalised recommendation does not interpret correctly the original guideline intention (or at least the actual user's interpretation is different) and the computer reminder is perceived as "erroneous", and indeed it is.

The approach of interacting with users about their non-compliance is conceptually similar to Perry Miller's proposal in his seminal work on medical critiquing systems in the eighties [61]. He proposed this type of system with the aim of enhancing the acceptance of conventional medical expert systems by physicians. In [62], a critiquing system is defined as a "decision support system that allows the user to make the decision first; the system then gives its advice when the user requests it or when the user's decision is out of the system's permissible range." Critiquing systems observe the inputs and decisions of the user and try to verify the decisions. Attracting attention only in critical situations, these systems are minimally intrusive and thus more acceptable. Thais is exactly the rationale underlying the development of RoMA. What really differentiates current systems from those developed 20 years ago is their integration within existing information systems, which of course has been facilitated by the dramatic technological progress achieved.

5.1. Compliance Verification based on Computational Logic

RoMA is a rule-based tool. Different approaches exist to the compliance checking task. For example, in [63] (chapter from this book), the authors propose GPROVE, a Computational Logic-based framework for verifying the compliance of careflow execution traces with respect to a formal careflow model. This framework provides a graphical language for the specification of careflow processes, as well as their mapping onto a logic-based formalism (namely, the SCIFF language), and a proof procedure for automatic compliance verification. The SCIFF language focuses on the concepts of 'happened' and (not) 'expected' events, and on the relations between them: such relations are expressed by means of rules, that allow the specification of the expected behaviour (in terms of expected or prohibited events), given that certain other events have happened. Each rule, named Social Integrity Constraint, has the form

body → head,

where body is a conjunction of events which have happened, and head is a disjunction of conjunction of expectations. Happened, expected and forbidden events (denoted respectively by the H, E and EN functors) are characterized by the description of the event itself and by the time instant the event has happened (or is expected to happen). Both the description of events and time instants can contain variables, which can be constrained. For example, the Social Integrity Constraint:

$H(executePAPTEST(Center, Pat), Te) \rightarrow E(sendResult(Center, Pat), Ts) \wedge Ts > Te \wedge Ts < Te + 15$

states that after an executePAPTEST event has occurred, a sendResult event is expected within 15 time units. The operational counterpart of the SCIFF language is an abductive proof procedure that automatically generates expectations, given the observed happened events and a set of integrity constraints (rules). Its most distinctive feature is the ability to check that the generated expectations are fulfilled by the actual participants behaviour, i.e., events expected to happen have actually happened, and forbidden events have not happened. The GPROVE framework has been successfully applied to the Cervical Cancer Screening Process implemented by the sanitary organization of the Emilia Romagna region of Italy.

6. Correlating Non-compliance to Clinical Data and Outcomes

To assess the impact of a guideline, it is interesting to check whether compliance is related to the health and/or economic outcomes. However, study design is not easy. As a matter of fact, the implementation of a clinical guideline could be considered as any other intervention affecting clinical practice, for example the introduction of a new drug, a new surgical procedure, etc.. It is well-known that the best way to assess the impact of such interventions is the randomized, controlled clinical trial. For guideline assessment, the "case" arm of the trial should be composed of patients treated accordingly to the guideline, and the "control" arm by patients treated traditionally. Nevertheless, with respect to a new drug evaluation, there is some difference:

- it is clearly impossible to perform a blind study and
- it is difficult to have a real “control” group, because we can’t be sure that physicians don’t know the guideline (guidelines are published, and their diffusion is promoted by several medical associations).

A weaker but suitable alternative to such a trial is a prospective, observational study, where patients' data are used to quantify the non-compliance with the guideline. One example is reported in [64] where a scale for measuring the non-compliance level for each patient was assessed. It is a numeric scale, that may range from 0 to 47 (the maximum number of guideline recommendations that may be violated for a single patient), and it has been used together with other variables in uni- and multivariate statistical models. This means that there are not two “a priori” different groups of patients, one treated according to the guideline and one not: the compliance with guidelines is measured “ex-post”.

The scale has been used as a numeric variable in a regression model (proportional hazards Cox model), where the dependent variable was the survival at 6 months follow-up. By means of multivariate analysis, it is possible to correct for multiple possible confounding factors. The fact that the number of non-compliances still remained statistically significant after correction for age, severity, comorbidities, etc. is a strong indication of its importance.

In principle, the trial methodology could be profitably applied to assess the benefit of a guideline implementation method: for example, to test whether computerised decision support is effective, the patients' outcome of Stroke Units supported by a computerised system could be compared with that of Stroke Units where only paper-based guidelines are available. However, still some problems remain with the true possibility of randomising Stroke Units, because decision support systems require technological facilities that cannot be easily implemented everywhere.

In appreciating the benefit of a guideline there are also specific studies on the effectiveness of single recommendation. For example, if a recommendation is intended to decrease the recurrence of a disease, the proportion of relapses could be compared in patients treated according or not according to that recommendation. Often these intermediate outcomes (other examples are the number of complications, their type, the length of hospital stay, etc.) are easier to measure than the main outcomes, such as survival, while being strongly correlated with it.

6.1. Compliance Versus Patients' Characteristics

The case study in Paragraph 3.1 discussed correlation between guideline compliance and surgeons' characteristics. Similarly, it is possible to investigate whether there are features of patients which affect physicians' behaviour. Maviglia et al. [65] report on rate of compliance with NCEP cholesterol guidelines: patient-specific factors associated with compliance included being male (37% compliance vs 24%) and being white (34% vs 26%). Decisions on patients over 79 and under 50 years old were also less likely to be compliant (22% vs 34% for 50–79 year olds).

A similar result was obtained for the SPREAD guideline compliance. Looking at the distribution of non-compliance in the subacute phase of the stroke, two modes were noted. Investigating the composition of the two populations corresponding to the bimodal distribution in terms of patients' characteristics, it was found that the right-most mode (the one with more non-compliance) was mainly composed of elderly people. This phenomenon is due to a well-known cultural bias: stroke in old patients is thought to be too debilitating, leading physicians to a less aggressive care, even though there is no evidence that elderly people do not benefit from those guidelines' recommendations.

Interesting analytical approaches have been proposed to investigate non-compliance in the light of patients' data. Svatek and Razavi [66,67] adopt a data-mining approach. In [66] information on frequently occurring non–compliance patterns is associated with other patterns in patient data. The inferred association rules look promising as source of explanatory hypotheses for physicians. Svatek's experiment was carried out on 48 data records of patients treated for hypertension. In [67] data from breast cancer patients admitted to Linköping University Hospital between 1990 and 2000 were analyzed. Considering cases that were not treated according to guidelines, four treatment rules were derived by decision tree induction. Comparing these rules with guideline recommendations allows alert to be generated when inconsistencies with the guidelines may appear.

7. Research Agenda

There are still several open issues in the area of computer-support to compliance with clinical practice guidelines. First of all, it is true that information technology can improve compliance, through real-time generation of patient-tailored recommendations, but the presentation modality of recommendations to users ha not been deeply studied yet. We are not only referring to the classical GUI (graphical user interface), but also to the physical device where the user must be alerted. Smart combination of synchronous and asynchronous communication (including interactive voice systems, mails, sms, mms, etc) must be in the research agenda, because timing and unobtrusiveness are fundamental for fostering compliance and avoid, on the contrary, annoying the users.

A second item in the agenda is the design and development of so-called *socio-technical* systems that take into account the whole guideline process: we have discussed, in the chapter, the importance of collecting the motivations for non-compliance. Once these motivations have been collected, someone must analyse them, and take corrective actions. Actions could be taken in the direction of foster compliance, but also in the

direction of updating the guidelines. That's why all the actors and organisational roles potentially involved should be considered since the beginning. Some research in this area could be drawn on the community of practice theories [68].
A third issue concerns formal models for compliance evaluation. As described along the book, within the computerised guideline community, workflow models, Petri nets-based formalisation, computational logic and other formal models are being tested. However, further research is needed to make these models flexible enough to face the dramatically large number of exceptions that can arise in a patient's clinical path.

Acknowledgements

The author thanks Silvia Panzarasa for her invaluable support in the design and technical implementation of the system for the Stroke Unit, Mario Stefanelli for fruitful discussion, and Anna Cavallini and Giuseppe Micieli, who provided medical knowledge and support computerised guideline implementation in their hospital units.

References

[1] R.N. Greenberg, “Overview of patient compliance with medication dosing: A literature review”. Clinical Therapeutics 1984, 6(5), 592-599

[2] M.R. DiMatteo, “Variations in patients’ adherence to medical recommendations: A qualitative review of 50 years of research”, Medical Care 2004, 42(3), 197-209

[3] L.C. Walker, M.O. Hogstel, P. Burns, “Teaching older adults to self-manage medications: Preventing adverse drug reactions”, Journal of Gerontological Nursing 2005, 31(4), 32-42

[4] J. Bouaud, B. Seroussi, E.C. Antoine, L. Zeliek, M. Spielmann, Reusing OncoDoc, a guideline-based decision support system, across institutions: a successful experiment in sharing medical knowledge, Proceedings AMIA 2000

[5] D.B. Fridsma, J.H. Gennari, M.A. Musen, Making Generic Guidelines Site-Specific, Proceedings AMIA Annu Fall Symposium 1996, 597-601

[6] A. Advani, Y. Shahar, M.A. Musen, Medical Quality Assessment by Scoring Adherence to Guideline Intentions, Journal of American Medical Informatics Association 2002 Nov–Dec; 9(6 Suppl 1)

[7] K. Pyorala, S. Lehto, D. De Bacquer, J. De Sutter, S. Sans, U. Keil, D. Wood, G. De Backer, EUROASPIRE I Group; EUROASPIRE II Group. Risk factor management in diabetic and non-diabetic patients with coronary heart disease. Findings from the EUROASPIRE I AND II surveys. Diabetologia. 2004,47(7), 1257-1265

[8] P. Ciccarese, E. Caffi, S. Quaglini, M. Stefanelli, Architectures and Tools for innovative Health Information Systems: the Guide Project. International Journal of Medical Informatics , M. Fieschi, M. Stefanelli and C. A. Kulikowski, ed. 74, 553-562

[9] E. Van Ganse, L. Laforest, E. Alemao, G. Davies, S. Gutkin, D. Yin, Lipid-modifying therapy and attainment of cholesterol goals in Europe: the Return on Expenditure Achieved for Lipid Therapy (REALITY) study. Current Medical Research and Opinion 2005 Sep;21(9),1389-1399

[10] T. Hockley, M.A. Gemmil, A Policy Analysis Centre Study on Cholesterol Guidelines. European Cholesterol Guidelines Report. *European Atherosclerosis Society (EAS)* di Helsinki, June 2007

[11] E.C. Geehr, R.F. Salluzzo. Clinical practice guidelines: promise or illusion? Physician Exec. 1990 Jul-Aug;16(4),13-6

[12] R. Grilli, J. Lomas. Evaluating the message: the relationship between adherence rate and the subject of a practice guideline. Medical Care 1994,32(3), 202-213

[13] M.D. Cabana, C.S. Rand, N.R. Powe, Why don't physicians follow clinical practice guidelines? a framework for improvement [review]. Journal of the American Medical Association 1999,282,1458-65

[14] M. Eccles, E. McColl, N. Steen, N. Rousseau, J. Grimshaw, D. Parkin, I. Purves, Effect of computerized evidence based guidelines on management of asthma and angina in adults in primary care: cluster randomised controlled trial, British Medical Journal 2002, 325

[15] L.L.Leape, J.S.Weissman, E.C. Schneider, R.N.Piana, C.Gatsonis, A.M. Epstein, Adherence to practice guidelines: the role of the speciality society guidelines, American Heart Journal 2003,145,19-26

[16] E.S. Barner, S. Baker, E. Funkhouser, Do local opinion leaders augment hospital quality improvement efforts? A randomized trial to promote adherence to unstable angina guidelines. Medical Care 2003, 41, 420-431

[17] P. Jami, P. Smith, S. Moningi, V. Moningi, S.A. Martin, G. Rosencrance, B.J. Reyes, Compliance with Joint National Committee 7 guidelines in hypertension management in a teaching institution. American Journal of Medical Quality, 2007 Jul-Aug;22(4),251-258

[18] I. Durand-Zaleski, C. Colin, C. Blum-Boisgard, An attempt to save money by using mandatory practice guidelines in France Education and debate, British Medical Journal 1997,315, 943-946

[19] P. Durieux, C. Chaix-Couturier, I. Durand-Zaleski, P. Ravaud, From clinical recommendations to mandatory practice. The introduction of regulatory practice guidelines in the French healthcare system. International Journal of Technology Assessment in Health Care. 2000,16(4),969-975

[20] B. Hurwitz, Clinical guidelines: Legal and political considerations of clinical practice guidelines. British Medical Journal 1999, 318, 661-664

[21] S.E. Carrick, B. Bonevski, S. Redman, J. Simpson, R.W. Sanson-Fisher, F. Webster, Surgeons' opinions about the NHMRC clinical practice guidelines for the management of early breast cancer. Medical Journal of Australia 1998, 169, 300-305.

[22] J.E. Pelly, L. Newby, F. Tito, S. Redman, A.M. Adrian, Clinical practice guidelines before the law: sword or shield?, The Medical Journal of Australia 1998, 169, 330-333

[23] S. Birch, A. Gafni, On being NICE in the UK: guidelines for technology appraisal for the NHS in England and Wales, Health Economics 2002 Apr,11(3),185-191

[24] D.F. Lobach, W.E. Hammond, Computerized decision support based on a clinical practice guideline improves compliance with care standards. American Journal of Medicine 1997 Jan,102(1),89-98

[25] K. Kawamoto, C.A. Houlihan, E.A. Balas, D.F. Lobach, Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. British Medical Journal 2005,330,765

[26] G.P. Purcell, What makes a good clinical decision support system, Editorial. British Medical Journal 2005 330, 740-741

[27] K.M Albert, Integrating knowledge-based resources into the electronic health record: history, current status, and role of librarians, Medical reference services quarterly 2007 Fall,26(3),1-19

[28] R.L. Engle Jr., Attempts to use computers as diagnostic aids in medical decision making: a thirty-year experience. Perspectives in biology and medicine 1992 Winter,35(2),207-219

[29] E. Coiera, Putting the technical back into socio-technical systems research, International Journal of Medical Informatics 2007 Jun,76 Suppl 1,98-103

[30] I. Sim, P. Gorman, R.A. Greenes, R.B. Haynes, Kaplan, H. Lehmann, P.C. Tang, Clinical Decision Support Systems for the Practice of Evidence-based Medicine, Journal of the American Medical Informatics Association 2001,8,527-534

[31] S. Panzarasa, S. Quaglini, G. Micieli, S. Marcheselli, M. Pessina, C. Pernice, A. Cavallini, M. Stefanelli, Improving compliance to guidelines through workflow technology:implementation and results in a Stroke Unit. Medinfo 2007 Proceedings

[32] M. A. Gilligan, J. Neuner, R. Sparapani, P. W. Laud, A. B. Nattinger. Surgeon Characteristics and Variations in Treatment for Early-Stage Breast Cancer. Arch Surg. 2007,142,17-22

[33] Early stage breast cancer. Consens Statement. 1990 Jun 18-21,8(6),1-19. Review

[34] A.G. Ellrodt, Conner L, Riedinger M, Weingarten S. Measuring and Improving Physician Compliance with Clinical Practice Guidelines: A Controlled Interventional Trial. Annals of internal medicine (1995) 122, 4, 277-282

[35] R. Grol, Successes and failures in the implementation of evidence-based guidelines for clinical practice. Medical Care, 2001 Aug,39(8 Suppl 2),II46-54

[36] B.M. Farr.Reasons for noncompliance with infection control guidelines. Infect Control Hosp Epidemiol. 2000 Jun,21(6),411-416

[37] F. Eisinger, I. Ronda, B. Puig, J. Camerlo, M.H. Giovannini, V.J. Bardou, Breast cancer guidelines--Physicians' intentions and behaviors.. International Journal of Cancer. 2007 Mar 1,120(5),1136-40

[38] C.M. Farquhar, E. W. Kofa, J.R. Slutsky, Clinicians' attitudes to clinical practice guidelines: a systematic review. MJA 2002, 177, 502–506

[39] J.P. Campbell, V.A. Maxey, W.A. Watson, Hawthorne Effect: Implications for Prehospital Research. Annals of Emergency Medicine 1995, 26(5), 590-594

[40] S. Grufferman, Complexity and the Hawthorne Effect in Community Trials. Epidemiology 1999,10(3), 209-210

[41] A.L. Dans, L.F. Dans, G.H. Guyatt, S. Richardson, Users' guides to the medical literature:XIV. How to decide on the applicability of clinical trial results to your patient. Evidence-Based Medicine Working Group. Journal of the American Medical Association 1998 Feb18,279(7),545-549

[42] D. Inzitari, G. Carlucci, Italian Stroke Guidelines (SPREAD): evidence and clinical practice. Neurological Science 2006 Jun,27 Suppl 3,S225-7. Review

[43] D. Eddy, Doctors, Economics and Clinical Practice Guidelines: Can they be brought together? Office of Health Economics OHE publications, 1999

[44] Setting Priorities for Clinical Practice Guidelines. Institute of Medicine (IOM). The National Academic Press 1995, 57-72

[45] T. Delmothe, Wanted: guidelines that doctors will follow. British Medical Journal 1998, 307, 218

[46] P.J. Greco, J.M. Eisenberg, Changing physicians' practices. New England Journal of Medicine 1993, 329,1271-1273

[47] S. Quaglini, A. Cavallini, S. Gerzeli, G. Micieli, GLADIS Study Group (Guideline Application for the Decision making in Ischemic Stroke). Economic benefit from clinical practice guideline compliance in stroke patient management. Health Policy 2004,69(3),305-315

[48] P. Clayton, G. Hripsak, Decision support in haltcare. International Journal of Biomedical Computing 1995, 39, 59-66

[49] R.N. Shiffman, B.T. Karras, A. Agrawal, R. Chen, L. Marenco, S. Nath, GEM: A proposal for a more comprehensive guideline document model using XML. Journal of the American Medical Informatics Association 2000,7,488-98

[50] S.W. Tu, J.R. Campbell, J. Glasgow, M.A. Nyman, R. McClure, J. McClay, C. Parker, K.M. Hrabak, D. Berg, T. Weida,J.G.Mansfield,M.A. Musen,RM. Abarbanel,The SAGE Guideline Model: achievements and overview. Journal of the American Medical Informatics Association 2007,14(5),589-598

[51] Y. Shahar, S. Miksch, P. Johnson, An intention-based language for representing clinical guidelines. Proc AMIA Annu Fall Symposium 1996,592-596

[52] S. Panzarasa, S. Quaglini, A. Cavallini, S. Marcheselli,M. Stefanelli, and G. Micieli. Computerised Guidelines Implementation: Obtaining Feedback for Revision of Guidelines, Clinical Data Model and Data Flow AIME 2007, LNAI 4594 Proceedings ,ed R. Bellazzi, A. Abu-Hanna, and J. Hunter (Eds.), 461 - 466

[53] Canadian Task Force on the Periodic Health Examination: The periodic health examination. Canadian Medical Association Journal 1979,121,1193-1254.

[54] D.L. Sackett Rules of evidence and clinical recommendations on use of antithrombotic agents. Chest 1986 Feb, 89 (2 suppl.):2S-3S

[55] D.J. Cook, G.H. Guyatt, A. Laupacis, D.L. Sackett, R.J. Goldberg, Clinical recommendations using levels of evidence for antithrombotic agents. Chest 1995 Oct, 108(4 Suppl),227S-230S.

[56] S. Yusuf, J.A. Cairns, A.J. Camm, E.L. Fallen, B.J. Gersh, Evidence-Based Cardiology. London: British Medical Journal Publishing Group, 1998

[57] D. Atkins, M. Eccles, S. Flottorp , G.H. Guyatt, D. Henry, S. Hill , A. Liberati, D. O'Connell, A.D. Oxman, B. Phillips, H. Schünemann, T. Tan-Torres Edejer, G.E. Vist, J.W. Williams Jr and The GRADE Working Group. Systems for grading the quality of evidence and the strength of recommendations I: Critical appraisal of existing approaches BMC Health Service Research 2004, 4, 38

[58] R. Harbour, J. Miller, A new system for grading recommendations in evidence based guidelines. British Medical Journal 2001,323,334-336

[59] M.C. Weinstein, Clinical Decision Analysis, Saunders Company, 1980

[60] T.K. Gandhi, T.D. Sequist, E.G. Poon, A.S. Karson, H. Murff, D.G. Fairchild, G.J. Kuperman, D.W. Bates, Primary care clinician attitudes towards electronic clinical reminders and clinical practice guidelines. AMIA Annu Symp Proc. 2003,848

[61] P. Miller, (1986), Expert Critiquing Systems - Practise-Based Consultation by Computer, Springer

[62] J. van Bemmel, and Musen, M. (1997): Handbook of Medical Informatics, Springer.

[63] F. Chesani, E. Lamma, P. Mello, M. Montali, S. Storari, P. Baldazzi, M. Manfredi, Computer-based medical guidelines and protocols: A primer and current trends. Chapter Compliance checking of cancer-screening careflows: an approach based on computational logic, IOS Press, 2008.

[64] G. Micieli, A.Cavallini, S. Quaglini, Guideline Compliance Improves Stroke Outcome- A Preliminary Study in 4 Districts in the Italian Region of Lombardia. Stroke 2002, 33,1341-1347

[65] S.M. Maviglia, J.M. Teich, J. Fiskio, D.W. Bates, Using an Electronic Medical Record to Identify Opportunities to Improve Compliance with Cholesterol Guidelines. . Journal of General Internal Medicine 2001 August, 16(8), 531–537

[66] V. Svatek, A. Riha, J.Peleska, J.Rauch, Analysis of guideline compliance – a data mining approach. In: Symposium on Computerized Guidelines and Protocols (CGP-04), Praha 2004. IOS Press

[67] A.R. Razavi, H. Gill, H. Ahlfeldt, N. Shahsavar, A Data Mining Approach to Analyze Non-compliance with a Guideline for the Treatment of Breast Cancer. Studies in health technology and informatics 2007,129,591-595

[68] E. Wenger , R. McDermott, W.M Snyder. Cultivating Communities of Practice, HBS press 2002

This page intentionally left blank

Part II

Current Trends

This page intentionally left blank

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-183

Compliance Checking of Cancer-Screening CareFlows: an Approach based on Computational Logic

Federico CHESANI [a], Evelina LAMMA [b], Paola MELLO [a], Marco MONTALI [a], Sergio STORARI [b,1], Paola BALDAZZI [c], and Marilena MANFREDI [c]

[a] *DEIS, University of Bologna, Italy*
[b] *ENDIF, University of Ferrara, Italy*
[c] *Department of Public Health - Sanitary Agency of Bologna, Italy*

Abstract. Clinical guidelines and Careflow systems have been recently identified as a means to improve and standardize health care services. A number of ICT-based management solutions have been proposed, focussing on several aspects such as specification, process logs verification with respect to specification (compliance), enactment and administration of careflows.

In this paper we introduce the GPROVE framework, based on Computational Logic, and focused on the (formal) specification of careflows and on the compliance verification of the process executions w.r.t. the specified models. In particular, we show its application to the Cancer Screening Guideline used by the sanitary organization of the Emilia Romagna region, discussing its formalization in GPROVE and the results of the compliance checking applied to logs of the screening process.

Keywords. Careflow Monitoring System, Clinical Practice Guidelines, Compliance Verification, Computational Logic.

Introduction

Clinical decisions in modern health care organizations are progressively taken on evidence-based care [1]. To this end, the use of clinical practice guidelines is considered to be a fundamental step towards high quality and standardized health services. As reported in [2], clinical guidelines describe the activities of a medical team in a comprehensive manner, with the purpose of defining best practices for patient management. In particular, clinical guidelines describe also the behavioural aspects of medical work, i.e., the clinician's workflow (namely *careflow*). Careflows are individual case-based and involve the coordinated execution of multiple medical tasks performed by different health care subjects on a specific patient.

The adoption of computer-based guidelines and Careflow Management Systems (CfMS) is an important issue, especially when the health care services are provided through complex care processes which involve several health care professionals. The

[1] Corresponding Author: Sergio Storari, Dipartimento di Ingegneria, Università di Ferrara, via Saragat 1, 44100 – Ferrara, Italy; E-mail: sergio.storari@unife.it.

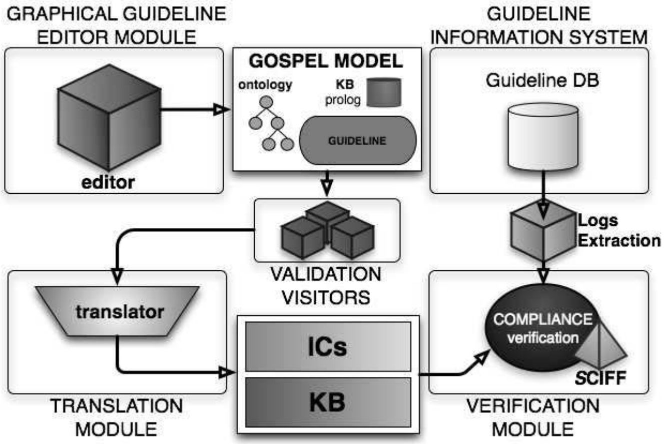


Figure 1. Architecture and modules of the GPROVE framework

goal of a CfMS is to "handle patients", in the sense of executing medical tasks in a specific order, as effectively and efficiently as possible. This management is made through the automated coordination, control, and distribution of tasks required to satisfy a given careflow process. A CfMS then is a set of tools which support careflow design (including its formal representation), enactment, administration, and monitoring.

In this work, we describe the Guideline PRocess cOnformance VErification framework (GPROVE, [3]), and its application to the Cancer Screening Guidelines adopted by the sanitary organization of the Emilia Romagna region (Italy). GPROVE is a set of tools for the specification and a-posteriori verification of the careflow process executions, comprehending: a graphical process definition language (GOSpeL); an automatic mapping/translation towards a formal language ($\mathcal{S}$CIFF); an operational counterpart (a proof procedure) of the $\mathcal{S}$CIFF formalism, that is used to verify the compliance of a given execution trace w.r.t. the defined careflow process. The verification step can be used in order to identify possibly undesired behaviours, as well as to analyze and comprehend features and characteristics of the real process that are not properly represented in the modeled process: if this is the case, log traces provide a feedback to the model, possibly leading to an improvement of the model itself.

1. Careflow Compliance Checking: the GPROVE Framework

The GPROVE framework [3] allows the user to specify a guideline and then to reason about the compliance of the observed behaviour. From the architectural viewpoint GPROVE is composed of several modules, as shown in Fig. 1.

The user specifies the medical guideline by means of GOSpeL (Guideline prOcess Specification Language, [3]), a graphical language inspired by flow charts that allows the representation of the *activities* and the *flows* of activity executions. The GOSpeL representation of the guideline is then translated into a formal specification based on the $\mathcal{S}$CIFF language [4]. Such translation is automatically performed by means of an algorithm [5], that "explores" the GOSpeL model and generates a set of *rules* (Integrity Constraints, ICs for short, and a knowledge base, SOKB for short) representing how

Table 1. GOSpeL graphical elements

family	type	notation	description
Activities	atomic activity	□	single atomic unit of work within the guideline
	complex activity	▣	Non-atomic unit of work. It encapsulates a new (sub)process definition.
	iteration	⊚	For-like cyclical complex activity
	while	⟐	While-like cyclical complex activity
Gateways	exclusive choice	◈	Data-based choice; each outgoing relation is associated to a logical guard.
	deferred choice	◇	Non-deterministic choice, without explicit logical conditions.
	parallel fork	⏢	Spanning of multiple execution threads
	parallel join	⏢	Synchronization of multiple threads of control
Start/ Completion Blocks	start	▶	Start point of a complex activity
	cyclic start	▶	Start point of a cyclical complex activity
	completion	◎	Completion point of a complex activity
	abort	◉	Abort the entire guideline

the guideline flow should be executed. Finally, a verification module takes as input the formal specification of the guideline and a log of relevant events. The compliance of such logs w.r.t. the given model is checked by the $\mathcal{S}$CIFF Proof Procedure [4], that notifies violations of the logged events w.r.t. the expected behaviour (as specified by the model).

1.1. The GOSpeL Careflow Specification Language

Like a typical flow-chart language, GOSpeL describes the guideline evolution using *blocks* and *relations* between blocks. These blocks are grouped into three families (as shown in Table 1):

activities, blocks which represent guideline activities at the desired abstraction level;
gateways, blocks used to manage the convergence and the divergence of control flow;
start and end, start and end points of (sub)processes.

Relations represent causal binary connections between blocks, expressing that the source block will be performed always before the destination one. Moreover, order relations show how the flow navigates through blocks, imposing a partial ordering among them.

A *simple* activity is a single atomic working step within the guideline: it models a situation where a guideline participant should perform something. *Complex* activities encapsulate new sub-processes definitions or repetitions of activities, and at their specification level they are managed like atomic activities. Gateways are used for modeling complex guidelines as long as they express workflow's decision points and activities concurrence. Decision points can be deterministic, or can be deferred choices. Concurrent activities definitions instead are supported by means of the *parallel fork* and *join* blocks.

To represent domain-related knowledge, GOSpeL adopts an ontology-based approach: the user can define two taxonomies, one for modeling activities at the desired abstraction level; a second taxonomy is used to describe domain's entities, namely actors, objects and terms. Each atomic activity block is semantically specified by mapping it onto an ontological activity and a set of participants. Creation and management of ontologies is performed through the Protégé [6] tool.

1.2. Formal Representation of the GOSpeL Model and Execution Traces Verification

The GOSpeL graphical model is translated into a simplified version of the formal language proposed by Alberti et al. in the SOCS European project for the specification and verification of interaction protocols (see [4] for a complete description). The social participant behaviour is represented by a set of (ground) facts called *happened events* and denoted by the functor $\mathbf{H}$ (that stands for "happened"). For example, $\mathbf{H}$*(cr_ScreeningInvitation(strsrg, center1, lab1, '10-02-06'), 7)* represents the fact that *center1* has sent the screening invitation to *strsrg* at time *7*. Future, desirable behaviour of the participants is represented by means of *expectations* about events that *should/should not* happen. Expectations have the same format as events, but they will, typically, contain variables to indicate that expected events are not completely specified. CLP constraints [7] can be imposed on variables to restrict their domain, e.g., for specifying temporal deadlines. Expectations about events that should happen are also called positive expectations, and are denoted by the functor $\mathbf{E}$; expectations about events that should not happen are named negative expectations, and are indicated by the functor $\mathbf{EN}$.

The way new expectations are generated, given the happened events and the current expectations, is specified by means of *Social Integrity Constraints* (ICs). An ICs has the form of *body* $\rightarrow$ *head*, expressing that when *body* becomes true then it is supposed that the events specified in *head* will happen. In this way, we are able to define guidelines as sets of forward (or backward) rules, relating happened events (in the *body*) to expectations (in the *head*). Moreover, it is possible to insert in the *head* also special predicates (*abducibles*, with functor *ABD*), typically used for hypothetic reasoning (in this work adopted to signal special situations).

The compliance verification is made by the operational counterpart of ICs, an abductive proof procedure named $\mathcal{S}$CIFF [8]. Given the partial or the complete history of a specific execution (*i.e.*, the set of already happened events recorded in an event log), this proof procedure generates expectations about participants behaviour so as to comply with ICs. The most distinctive feature of $\mathcal{S}$CIFF, however, is the ability to check that the generated expectations are *fulfilled* by the actual participants behaviour (i.e., that events expected to happen have actually happened, and forbidden events have not happened). If a participant does not behave as expected w.r.t. the model, the proof procedure detects and raises a violation. In particular, the proof, analyzing a log representing the activities performed during the careflow execution, can detect two kinds of guideline compliance violations: the first when an expected event is not found in the log ($\mathbf{E}$(Event) without the relative $\mathbf{H}$(Event)); the second when a prohibited event is found in the log ($\mathbf{EN}$(Event) with a corresponding $\mathbf{H}$(Event)).

2. The Cervical Cancer Screening Guideline

We have applied the GPROVE tools to a real case, the Cervical Cancer Screening Guideline proposed by the sanitary organization of the Emilia Romagna region of Italy [9].

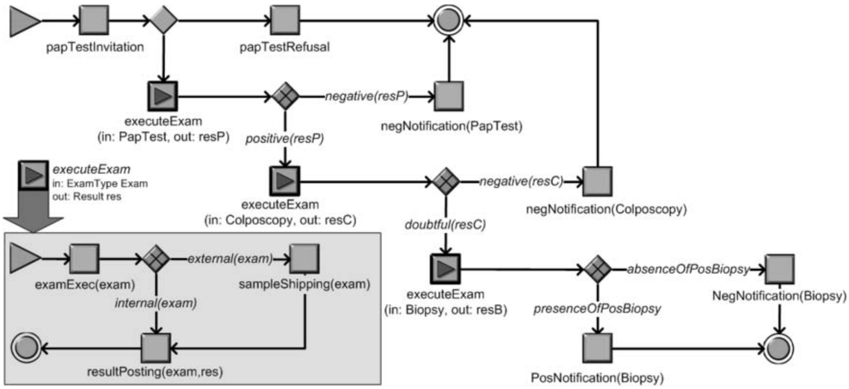


Figure 2. The GOSpeL representation of the example guideline

Cervical cancer is a disease in which malignant (cancer) cells grow in the tissues of the cervix. The screening program proposes several tests in order to early detect and treat cervical cancer, and it is usually organized in six phases: Screening Planning, Invitation Management, First Level Test (PAP-test), Second Level Test (Colposcopy), Third Level Test (Biopsy), and Therapeutic Treatment. Every participant is asked to repeat the screening process periodically (a round being thus three years long).

The workflow modeled by the Cervical Cancer Screening Guideline is quite complex and involves more than 50 activities performed by 15 different health care professionals and structures. For the sake of clarity we discuss here the formalization of a portion of the guideline (named in the paper $ExCervScreening$). $ExCervScreening$ prescribes that the screening process starts when the screening center sends a PAP-test invitation to a patient. The patient can decide to refuse or to go to the PAP-test center. In the latter case, the PAP-test is executed and the collected biological sample is sent to the anatomopathology laboratory. A report is sent back to the screening center, reporting a positive classification (cancer evidence found) or a negative classification (normal). In case of a positive report, the patient is invited for a Colposcopy; if positive again the patient is invited for a Biopsy and possibly for cancer treatment. Every time a negative response is reported instead, a letter is sent to the patient and a new screening round is scheduled.

2.1. Formalization of the Guideline into GOSpeL

The first step when formalizing a careflow consists of analyzing the specific domain, and to identify the actors and the activities that more characterize the process. These actors interact with each other by means of activities that involve documents, biological samples, etc. These elements are the objects exchanged during the guideline execution. E.g., the patient, the screening center, the PAP-test center and various laboratories can be identified as *actors*; the invitation letter, the biological samples and the notification letters are *objects*; while the posting of an invitation, the execution of an exam, as well as the shipping of the samples are *activities* performed during the execution of the screening.

In Figure 2 the GOSpeL translation of the *ExCervScreening* is shown. The screening process starts with the activity block *papTestInvitation*, followed by a deferred choice that models the decision taken by the patient (*papTestRefusal* activity block, or *executeExam* macro block, instantiated on *PAP-Test*).

The *executeExam* macro block is used to represent a set of activities that are common to several exams. In particular, the PAP-test, the Colposcopy, and the Biopsy share the *examExec* and the *resultPosting* activities. Moreover, since the PAP-test and the Biopsy samples are analyzed in different labs than the execution ones, they are considered "external" and require a *sampleShipping* activity. The GOSpeL definition of the *executeExam* macro block is also shown in Figure 2: it takes as input the type of the exam, and returns as output the exam result. An exclusive choice is performed, depending on the exam type, and one among two different paths is selected (corresponding to the external/internal exam type). The criteria for such a choice can be defined in a Prolog knowledge base associated to the GOSpeL diagram.

In case the PAP-test is positive[2], the diagram specifies the execution of the Colposcopy (*execExam* with "colposcopy" as exam type). Again, the result (*resC* in this case) is tested and, if positive or doubtful, a further exam (Biopsy) is performed. Finally, the Biopsy macro block is followed by an exclusive choice used to distinguish if at least one sample has been found positive or not. A notification to the patient follows.

2.2. Formal Representation of the Screening Guideline

The GOSpeL diagram shown in Figure 2 has been automatically translated into a set of $\mathcal{S}$CIFF ICs, providing a formal representation. The algorithm [5] makes a partition of the diagram into several sub-sets; then, for each sub-set, an Integrity Constraint is generated, taking into account the links between the different activities involved.

In Eq. 1 it is presented an Integrity Constraint obtained by the translation algorithm. In particular, the IC states that after an event *papTestInvitation* has occurred, either the exam is performed (positive expectation about the event *execExam*) and refusal is forbidden, or a *refuse* event is expected (and the execution of the test is prohibited).

$$\begin{aligned}
&\mathbf{H}(papTestInvitation(ScrCentre, Pat, Date, IdExam), T_{inv})\\
\rightarrow&\mathbf{E}(execExam(ExamCentre, Pat, papTest, IdExam, Date), T_{pap})\ \wedge\ T_{pap} > T_{inv}\\
&\wedge\,\mathbf{EN}(refuse(ScrCentre, Pat, Date), T_{ref})\ \wedge\ T_{ref} > T_{inv}\\
\vee&\mathbf{E}(refuse(ScrCentre, Pat, Date), T_{ref})\ \wedge\ T_{ref} > T_{inv}\\
&\wedge\,\mathbf{EN}(execExam(ExamCentre, Pat, papTest, IdExam, Date), T_{pap})\ \wedge\ T_{pap} > T_{inv}.
\end{aligned}\tag{1}$$

It is worthy to notice several aspects of the rule presented in Eq. 1. Firstly, the rule specifies for each event a set of parameters not present in Figure 2. Such list of parameters has been derived by the ontology definition of the activity *papTestInvitation*. Secondly, temporal constraints have been imposed over the variables T_{inv}, T_{pap} and T_{ref}, to define the desired temporal sequence of the events (i.e., a *refuse* activity should happen *after* the invitation for the test, and not before). Finally, negative expectations have been introduced to make exclusive the choice between attending the test and refusing: i.e., it is not possible that a patient *Pat* attends the PAP-test and also refuses to make it.

Eq. 2 and 3 instead show how the macro block *executeExam* has been represented. In particular, the choice between the shipping of the biological sample and the result notification is made by using the user-defined prolog predicates *external(X)* and *internal(X)*.

[2] To this end, two predicates positive(resP) and negative(resP) have been defined.

$$\begin{aligned}
&\mathbf{H}(execExam(ExamCentre, Pat, ExamType, IdExam, Date), T_{ex})\\
\rightarrow&\mathbf{E}(sampleShipping(ExamType, IdExam, Date), T_{ship}) \ \wedge\ T_{ship} > T_{ex}\\
&\wedge\ external(ExamType)\\
\vee&\mathbf{E}(resultPosting(ExamType, IdExam, Date, Res), T_{res}) \ \wedge\ T_{res} > T_{ex}\\
&\wedge\ internal(ExamType).
\end{aligned} \tag{2}$$

$$\begin{aligned}
&\mathbf{H}(sampleShipping(ExamType, IdExam, Date), T_{ship})\\
\rightarrow&\mathbf{E}(resultPosting(ExamType, IdExam, Date, Res), T_{res}) \ \wedge\ T_{res} > T_{ship}.
\end{aligned} \tag{3}$$

3. Compliance Verification

The *ExCervScreening* guideline, whose GOSpel diagram is shown in Figure 2, has been automatically translated into fourteen different ICs. These constraints have been provided as input to the SOCS-SI [8] tool for verifying the v of the logs.

In the case of the Emilia Romagna sanitary organization, all the cancer screening information are recorded in a database. A syntactical translation towards a more suitable and "log-like" representation has been applied to the data, and then the compliance checking has been performed. Let us consider for example a simple execution of the above careflow process, represented by a set of happened events:

1. **H**(*papTestInvitation(scr1,'00000260','1995-11-6','b_95017025'),5*)
2. **H**(*execExam(hosp1, '00000260', papTest, 'b_95017025','1995-11-6'),7*)
3. **H**(*sampleShipping(papTest,'b_95017025','1995-11-6'),20*)
4. **H**(*resultPosting(papTest,'b_95017025','1995-11-22', neg),30*)
5. **H**(*negativeNotification(scr1,'00000260',papTest,'b_95017025','1995-11-22', neg),30*)

This log represents a typical sequence of events: the patient *'00000260'* is invited to the screening, and she attends the first-level test (log entries 1 and 2); then the biological sample is sent to the laboratory, and the result is sent back to the screening center (log entries 3 and 4). Finally, a notification letter is sent to the patient (log entry 5).

Such a log is indeed compliance with the given careflow specification: each event triggers one or more Integrity Constraints, thus generating expectations about future events. E.g., log entry 1 triggers the rule shown in Eq. 1: an expectations about executing the exam is generated. Such expectation is fulfilled by the event traced in log entry 2. Suppose now that the *execExam* event would not be present in the log: then, the expectation about it would remain not fulfilled. The $\mathcal{S}$CIFF proof procedure then would try to check if the alternative behaviour specified in Eq. 1 could be satisfied instead: unfortunately, a *refuse* event is missing, and the $\mathcal{S}$CIFF proof procedure would raise a violation.

3.1. Verification of the Cancer Screening Logs

The compliance verification approach discussed in this paper has been applied to a database containing 1950 careflow executions. The careflow model has been specified by the authors, on the base of the cancer screening process [9].

In order to fully test our tools, some wrong behaviours have been introduced in this database. Each screening round has been checked as a single interaction (hence we did not check the compliance for the repetition of the screening rounds). Each screening contains several events: from the minimum of one (the screening invitation followed by no response) to the maximum of 18 (the whole careflow). The total time occurred to verify the compliance of the 1950 executions w.r.t. the careflow model was 12 minutes (average time of 369 msec. for each execution). 1091 executions resulted to be not compliance w.r.t. the formalization we have initially proposed. These results were analyzed by a screening expert which confirmed all the compliant classifications and proposed some changes to the careflow model: in fact, some traces classified as compliant by the domain expert were instead considered as non compliant w.r.t. the initial model. E.g., it was not taken into account that some patients, asked to participate to the screening, simply decide to not answer at all. Then Eq. 1 has been substituted by Eq. 4.

$$\begin{aligned}
&\mathbf{H}(papTestInvitation(ScrCentre, Pat, Date, IdExam), T_{inv})\\
&\rightarrow\mathbf{E}(execExam(ExamCentre, Pat, papTest, IdExam, Date), T_{pap})\ \wedge\ T_{pap} > T_{inv}\\
&\quad\wedge\,\mathbf{EN}(refuse(ScrCentre, Pat, Date), T_{ref})\ \wedge\ T_{ref} > T_{inv}\\
&\vee\mathbf{E}(refuse(ScrCentre, Pat, Date), T_{ref})\ \wedge\ T_{ref} > T_{inv}\\
&\quad\wedge\,\mathbf{EN}(execExam(ExamCentre, Pat, papTest, IdExam, Date), T_{pap})\ \wedge\ T_{pap} > T_{inv}\\
&\vee ABD(warning(inivitation\_not\_respected), T_{inv})\\
&\quad\wedge\,\mathbf{EN}(AnyEvent, T_{any})\ \wedge\ T_{any} > T_{inv}.
\end{aligned} \tag{4}$$

In Eq. 4, the possibility of abducing a *warning* predicate has been used to consider as compliant a log composed by the invitation event only (see [4] for more details on the $\mathcal{S}$CIFF abductive framework). Furthermore, any event after the invitation has been prohibited by using a negative expectation: hence, the logs considered as compliant by the modified diagram are those logs that contain only the invitation event and nothing else. Using this revised model, we avoided false non compliant classifications, reducing the number of executions classified as non compliant to 44: this result agrees indeed with the "wrong behaviour" executions we artificially introduced.

3.2. Verifying Particular Logs Features

The $\mathcal{S}$CIFF Proof Procedure and the SOCS-SI tools can be used also to verify particular features or situations that characterize a certain log. For example, it could be interesting to signal when certain situations happens, without raising a violation. This can be easily done by extending the formal representation of the careflow process, e.g., by adding new integrity constraints, or by modifying existing ones. To illustrate this capability, let us introduce some details about the invitation. Contextually to the invitation, a date for the PAP-test is automatically booked and proposed to the patient. The patient can refuse the proposed date, and phone directly to the screening center for booking another appointment. This process is not explicitly defined in the screening guideline, and the database does not store any information about it. The only information stored is about the invitation to the PAP-test, and the execution of the exam (together with the date it was executed). To understand how frequently this situation can happen, we added a new integrity constraint, shown in Eq. 5, to the formal specification.

$$
\begin{aligned}
&\mathbf{H}(papTestInvitation(ScrCentre, Pat, Date, IdExam), T_{inv})\\
&\wedge\ \mathbf{H}(execExam(ExamCentre, Pat, papTest, IdExam, DateExec), T_{pap})\\
&\wedge\ DateExec - Date > 15\\
\rightarrow & ABD(warning(delay\_higher\_than\_15\_days, Date, DateExec), T_{inv}).
\end{aligned}
\tag{5}
$$

Eq. 5 states that if a patient has been invited to attend the PAP-test, and the exam took place more than 15 days later the scheduled exam, a warning should be issued. We repeated the analysis of the logs, and we discovered that 200 times the PAP-test has been attended more than 15 days later w.r.t. the initial schedule. The delay could be explained by the fact that the screening center allocates in advance a certain number of slots: as a consequence, free slots for new booking are not immediately available.

4. Related Work and Conclusions

Several medical support systems have been proposed to represent and manage clinical guidelines, and some of them support also various verification tasks. In [10], compliance of the actual treatment of a specific patient is checked with the ideal behavior prescribed by the guideline by using temporal logic and model checking techniques. Two non-compliances are considered: non-compliant action ordering (i.e., prescribed actions are performed, but in a wrong order) and non-compliant actions (i.e., some guideline's actions cannot be prescribed at all for a specific patient). GPROVE is able to identify both the types of non-compliances; moreover, thanks to its computation logic roots, can also address other types of non-compliances such as, e.g., temporal deadlines verification. In fact, one of the main advantages of using a first order language such as computational logic w.r.t. more classical approaches based on temporal logics such as LTL or CTL, is the possibility of introducing variables and reasoning upon them. Hence, GPROVE naturally supports temporal constraints (intended as CLP constraints over a variable representing a time point), as well as constraints over terms containing variables.

Other approaches focus instead on the static verification of general and domain-dependent properties, aiming at identifying design errors and inconsistencies. A typical approach consist of mapping guidelines specification into suitable formal languages in order to provide automatic tools for verification. In [11] the authors propose the translation of GLARE [12] to the SPIN Model Checker, in order to discover inconsistencies in the model. In [13] a clinical guideline specified with Asbru [14] is viewed as a hierarchical plan and mapped onto the KIV interactive theorem prover. A variant of Interval Temporal Logic is then used to specify and verify properties.

In this work, we have introduced the GPROVE framework, showing its application to a real case, namely the Cancer Screening Guideline of the Emilia Romagna region. We have shown, by formalizing the guidelines in the logic-based formalism $\mathcal{S}$CIFF, how it is possible to perform the compliance checking of the execution traces w.r.t. the modeled careflow process. Such verification can be used to identify undesired behaviours, to comprehend features of the real process not highlighted by the model, and to identify strengths and weaknesses of the careflow process.

Future work will be devoted to complete the development of the GPROVE modules. Other graphical formalisms will be considered for the guideline specification, with the goal of providing a mapping to the $\mathcal{S}$CIFF language, and to support compliance checking also in the context of other frameworks. Future extensions will tackle run-time verification, static verification of properties, and enactment of guidelines models.

Acknowledgements

This work has been partially supported by NOEMALIFE under the “SPRING” regional PRRITT project, by the PRIN 2005 project “Specification and verification of agent interaction protocols” and by the FIRB project “TOCAI.IT”. We would like to thank reviewers and participants of the “ECAI 2006 Workshop on AI Techniques in Healthcare - Evidence-based Guidelines and Protocols” for the fruitful discussions on a preliminary version of this work. We would like to thank also Dr. Natalina Collina (Sanitary Agency of Bologna) for her contribution on the screening process management.

References

[1] G.J.A. Muir. *Evidence-based Healthcare*. Churchill Livingston, London, 1997.

[2] Careflow management systems. http://www.openclinical.org/briefingpaperStefanelli.html.

[3] F. Chesani, P. De Matteis, P. Mello, M. Montali, and S. Storari. A framework for defining and verifying clinical guidelines: A case study on cancer screening. In F. Esposito, Z. W. Ras, D. Malerba, and G. Semeraro, editors, *ISMIS*, volume 4203 of *LNCS*, pages 338–343. Springer, 2006.

[4] M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. Verifiable agent interaction in abductive logic programming: the SCIFF framework. *ACM Transactions on Computational Logics*. Accepted for publication.

[5] M. Montali F. Chesani, P. Mello and S. Storari. Testing careflow process execution conformance by translating a graphical language to computational logic. In Bellazzi et al. [15].

[6] Protegè. http://protege.stanford.edu/.

[7] J. Jaffar and M.J. Maher. Constraint logic programming: a survey. *JLP*, 19-20:503–582, 1994.

[8] The $\mathcal{S}CIFF$ abductive proof procedure. Available at http://lia.deis.unibo.it/Research/sciff/.

[9] Cervical Cancer Screening in Emilia Romagna (Italy). http://www.regione.emilia-romagna.it/screening/.

[10] P. Groot, A. Hommersom, P. J. F. Lucas, R. Serban, A. ten Teije, and F. van Harmelen. The role of model checking in critiquing based on clinical guidelines. In Bellazzi et al. [15], pages 411–420.

[11] L. Giordano, P. Terenziani, A. Bottrighi, S. Montani, and L. Donzella. Model checking for clinical guidelines: an agent-based approach. In *Proc. of the American Medical Informatics Association (AMIA) Symposium*, 2006.

[12] P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G. Molino, and G. Correndo. Applying artificial intelligence to clinical guidelines: The GLARE approach. In *AI*IA*, volume 101, pages 536–547, 2003.

[13] A. ten Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen, P. J. F. Lucas, S. Miksch, W. Reif, K. Rosenbrand, and A. Seyfang. Improving medical protocols by formal methods. *Artificial Intelligence in Medicine*, 36(3):193–209, 2006.

[14] Y. Shahar, S. Miksch, and P. Johnson. The asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artif. Intelligence in Medicine*, 14(1-2):29–51, 1998.

[15] R. Bellazzi, A. Abu-Hanna, and J. Hunter, editors. *11th Conf. on Artificial Intelligence in Medicine, AIME, Amsterdam, The Netherlands, July 7-11, 2007, Proc.*, volume 4594 of *LNCS*. Springer, 2007.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-193

Medical Guidelines for the Patient: Introducing the Life Assistance Protocols

David DOMÍNGUEZ [a,1], Carlos FERNÁNDEZ [a], Teresa MENEU [a],
Juan Bautista MOCHOLÍ [a], and Riccardo SERAFIN [b]
[a] *Institute ITACA, Universidad Politécnica de Valencia, Spain*
[b] *e-Services for Life and Health, San Raffaele Scientific Institute, Italy*

Abstract. This paper introduces our preliminary results in the modeling of Life Assistance Protocols, a new vision of medical guidelines and protocols through the lenses of p-Health. In this context the patientńs role in the process is emphasized, the actions to be performed less defined and not only clinical situations considered, but also healthier lifestyle promotion processes accounted for, where the personńs preferences and motivations play a key role.

We propose a complete framework, balancing on classical clinical guideline models and covering both the theoretical and the practical aspects of the problem, describing it from conceptualization to the execution environment.

Keywords. p-Health, Personalization, Life style, Workflows, Timed paralell automata

Introduction

One of the common issues that has become popular in recent years, in the domain of the e-Health technologies for citizen centered health systems, is the way to integrate disease prevention, control and treatment, into the person's daily life in a personalized and non-invasive way. This is one of the seeds that has led to the creation of a new concept: the Personal Health or p-Health.

The Life Assistance Protocol (LAP) model is defined within this context, and it is going to be the kernel of the framework described in this paper. A LAP is a set of guidelines, recommendations and prescriptions (actions) for a concrete need of the user, including not only health care actions but also healthier lifestyle related activities. In this setting user's needs may be pathologies (i.e. diabetes, heart failure), special conditions (i.e. pregnancy, elderliness), main health concerns (i.e. quitting smoking) or simply following a healthy lifestyle. The LAP defines, then, a set of actions in order to solve user needs. These actions represent the cycle (workflow) of the life of a person, in which each stage takes into consideration all the person dimensions: motivation, clinical status, personal context, etc. These dimensions are continuously re-evaluated to trigger moves forward and backward in the stages of the LAP.

[1] Corresponding Author: Institute ITACA, Universidad Politécnica de Valencia, Spain, email: dadotor@itaca.upv.es

Being presented in this paper is the creation of a complete framework that covers the LAP definition and representation, along with the final execution of the protocols actions via ICT systems.

The paper is organized as follows: in Section 1 we present the rationale that led us to the definition of the LAP model as an extension of existing clinical guidelines frameworks; Section 2 introduces the LAP Framework with particulare enphasis, in 2.1, on the definition of the the Conceptual Model and the illustration, through a hypothetical example, of how the elements fit together. The remaining of Section 2 describes the other components of the proposed framework, namely the Workflow and Execution Models. In Section 3 we discuss our plans for the implementation and evaluation of the framework, while in Section 4 we present our conclusions.

1. Rationale

Clinical guidelines (or *Care Plans*) are a powerful method for standardization and uniform improvement of the quality of medical care. Clinical guidelines are a set of schematic plans, at varying levels of abstraction and detail, for management of patients who have a particular clinical condition (e.g., insulin-dependent diabetes).

The application of clinical guidelines by care providers typically involves collecting and interpreting considerable amounts of data over time, applying standard therapeutic or diagnostic plans in an episodic fashion, and revising those plans when necessary [11]. Clinical Pathways differ from practice guidelines, protocols and algorithms as they are employed by a multidisciplinary team and have a focus on the quality and co-ordination of care [8].

However, in both cases the intervention of the patient in the execution of the protocols is very limited, and the protocols themselves do not include all the variability that patient daily life presents, which, at the end of the day, is one of the main causes of the non-compliance with the treatments when put into practice. LAP doesn't differ from the former definitions, in philosophy, but tries to empower those two dimensions, the patient and his environment, as active actors of the care plan outside hospitalized environment (not clinical) and the inclusion of the patient's daily activities and choices. LAP is, then, concerned on how the medical procedures are applied to meet the patient particular needs within the patient preference context. At the same time, as it has been mentioned, the LAP targets both those patients that suffer from chronic diseases but are in a stable status at home and do not need strong clinical interventions, and citizens at risk or wanting to carry out a healthier lifestyle (public health interventions).

In our vision, a solution that focuses on supporting the patients in such environments must take into account the following elements:

- The patient must be considered the main active actor in the care process, instead of the passive receiver of the actions performed by the caregivers;
- For the above reason, motivational aspects and possible non compliance must be accounted for in the core of the model, instead of being considered just as undesirable deviations from the normal path;
- In such a context, the recommendations and guidelines (actions) are usually less strict, in some cases even given without direct mediation of a health professional;

- On the other hand, the level of variability is greater and the need of adaptation to the patient's life (preference and context) much higher;

The framework we are proposing tries to address directly these needs and the peculiarities that such processes may present, and we believe that the management of such variability via ICT solutions is not only possible but also advisable, in order to improve patient's compliance and to give to all the involved actors the possibility to better adjust the guidelines to their needs, therefore increasing their efficacy.

2. LAP Framework

The classical approaches that can be found in literature related to the specification, representation and execution of medical guidelines are based on different models. They could be sorted as logic-based (PROforma), rule-based (Arden Syntax), Network-based (Prodigy), and Workflows as Petri Nets (Guide) [7]. The aim of these models is creating computer-interpretable guidelines that facilitate decision support, covering both computable data and clinical knowledge. There are also other models focused on the clinical knowledge representation by using XML guideline document models, such as GEM [12] or HGML [5], which in some cases can be also computable.

Most of the written guidelines implicitly define a workflow process, therefore by using a language definition based on workflows, and working in adapting the medical guidelines to it, they can be expressed in the mentioned language, and be decomposed in smaller tasks which can be easily executed under the figure of a coordinator. This model is basically focused on the workflow of the medical acts; where the actions to be performed are not taken into account at the beginning, and only the workflow's nodes are important. The evaluation of the indicators resulting from the execution of the actions in the nodes, will be used to obtain the jumping conditions from one node to another of the workflow.

All this drives us to create a complete framework, which must cover both the theoretical and the practical aspects of the problem. The framework will include the following set of models and tools:

- A Conceptual model;
- A Workflow model;
- An Execution model;
- A LAP's specification language (workflow-oriented);
- A set of software tools, such as a CASE and verification application based on graphical symbols to define workflows, a compiler to generate workflow LAP templates and a tool to adapt and personalize LAP templates with patient specific parameters (possibly taken from clinical records) in order to make a computable instance of the LAP;

The conceptual model will be presented in the next section and creates a theoretical base upon which all the framework stands. All the concepts in the model will be used in order to create templates, which will be filled by the domain experts through a graphical tool and instantiated by the system for a concrete person. This instance needs to be internally represented in a formal language that will be executed by the workflow model discussed in the following sections. These layers are represented in Figure 1.

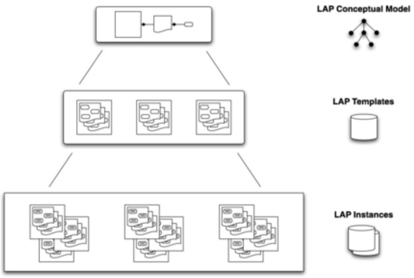


Figure 1. LAP layers from abstract to concrete

The final LAP instance will have such an amount of (qualitative and quantitative) information that will be easily computable.

In the following sections we introduce the workflow model specification, we introduce, briefly, the execution model and we discuss the specification languages used. As the LAP framework is still in its infancy, no support tools have been developed yet, apart for those described in the Execution Model which are needed to enact the specified workflows.

2.1. LAP Conceptual Model Definition

The LAP Conceptual Model can be defined as follows:

LAP: is a set of guidelines, recommendations and prescriptions for a concrete need of the user. It is defined by a general objective and it is composed of a set of nodes grouped into stages.

NODE: contains a set of actions[2], which are mutually independent, transactionally simultaneous and that will be executed over or by one or several determined actors. Each node will be defined by a description, a set of actions, indicators and reactions. A node could possibly include an explicit enumeration of user relevant data out of a predefined set (e.g. clinical record)[3]. A node is active if at least one action is in execution, otherwise it is inactive.

STAGE: is the set of active nodes in a concrete moment of time, and may have an associated label. There are two special types of stages to be defined here, initial stages (usually one per LAP) and objective stages, that can be more than one. The latter will be reached whether the general objective of the LAP has been achieved or the patient is in a stage in which the LAP is not able to continue (i.e. the person needs to go urgently to a hospital). However, these stages do not explicitly represent the status of the patient.

INDICATOR: is a function of time[4] and a (possibly empty) subset of the static or dynamic actor's features to a discrete set of values (indicator labels).

[2] All those actions related to a specific circumstance, like treating an edema.

[3] These might become inputs for actions or just be collected for human interpretation.

[4] Time is being considered to model timeout events that can trigger a reaction.

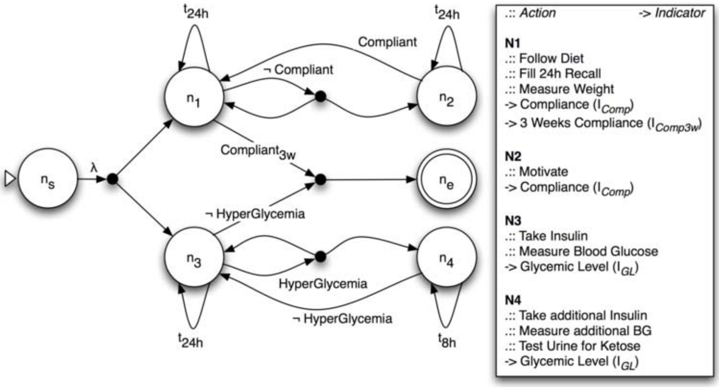


Figure 2. Graphical representation of simple LAP.

ACTION: is defined by an activity, an actor which executes the activity (executor) and one or more actors that benefit from the execution of the activity (addressee). The executor and the addressee can be the same actor.

ACTOR: is considered to be any user of the system or the system itself, and it is defined by a set of static or dynamic features, that might be retrieved from the person's clinical record, or even other sources containing non-clinical information, such as preferences, beliefs, habits, etc...

REACTION: models the transition between nodes, and it is defined as a function of one or more nodes and one ore more indicators labels to one or more nodes.

OBJECTIVE: is the desired outcome of the overall execution of the LAP. It is formally represented by a subset of the final nodes. When one of these nodes are reached by the patient it indicates that the LAP's objective has been achieved successfully.

In order to make the discussion as concrete as possible, we are now going to illustrate how this definition applies to a hypothetical example. The setting we consider is the following: a patient who recently discovered to suffer of diabetes, let's call him John, has finally managed to get used to his drug therapy. Now that he's able to control his glycemic level (GL) with drugs, his doctor wants him to go a step further and puts him on a diet. As soon as John is able to follow the diet regularly and reaches his target weight, the doctor will change the overall therapy relying more on the nutritional balance and reducing the drugs. To achieve these results, motivate and educate the patient to follow a diet, the doctor activates the LAP graphically presented in Figure 2.

The idea is that we have two main concurrent activities: trying to follow a diet and keeping the glycemic level controlled. For the former, when we detect a non compliance we keep on with the same activities but we add motivational support, which ends as soon as the person becomes compliant again. For the latter, we just increase the dosage of insulin and follow a more accurate monitoring process if hyperglycemia is detected.

This example is not realistic from a clinical point of view, for instance it lacks proper reactions if the user keeps being non-compliant or in hyperglycemia, notification to the doctor at LAP completion, etc. However, it is simple and yet meaningful enough to let us illustrate all the elements involved.

With respect to the definition above, this LAP is composed of six nodes $(n_s, n_1, \ldots, n_e)$. The possible activities that can be executed are things like measuring the weight, motivating the user providing him supporting information, filling a 24h recall questionnaire[5], etc. Notice that some of them will be actions where the executor and the addressee is the same (John, i.e. measuring the weight), while for others the executor and the addressee are different (the System and John, i.e. motivating).

We have two main indicators: a simple one related to the GL ($HyperGlycemia$ if $GL > 250$) and a more complex one related to the compliance in following the diet and filling the 24h recall ($Compliant$ if he reaches the target weight or continues to loose weight and he keeps on filling the 24h recall with results close to his diet caloric target)[6]. Moreover, we have timeouts that model the frequency at which actions have to be carried out in the different nodes[7].

The reactions can be derived directly from the graph but notice, for instance, than when a non compliance is detected in n_1 both n_1 and n_2 are concurrently activated.

Finally, having six nodes, potentially there exist 2^6 stages but due to the topology of the graph there exist only six possible stages: $q_0 = \{n_s\}$, $q_1 = \{n_1, n_3\}$, $q_2 = \{n_1, n_2, n_3\}$, $q_3 = \{n_1, n_3, n_4\}$, $q_4 = \{n_1, n_2, n_3, n_4\}$ and $q_5 = \{n_e\}$ which will be our objective stage.

Even with this very simple example it can be seen how this type of model helps in dealing with the multi-faceted dimensions defining a person health condition and how they correlate with each other. Through the separation of concerns, in our example the glycemic level and the diet compliance, we simplify the process definition but at the same time, through the power offered by stages, we are still able to capture those complex interactions.

2.2. LAP Workflow Model

As it has been mentioned in the previous section, the LAP framework includes a workflow model. The theoretical model that is going to be used for the LAP workflow is taken from the preliminary studies on Timed Parallel finite Automata (TPA) [3] which are based on Parallel Finite Automata (PFA). PFA are an improvement of Determinist Finite Automata (DFA) and model automata with capability to execute parallel activities.

A parallel automaton implies that more than one node is active at the same time -like in Petri Nets- but they also are supported by the concept of states, that approximates the PFA to the DFA by joining the parallel active nodes into a different state, and finally this, based on states concept model, is an abstraction that maps the PFA to a DFA which is a well controlled framework. TPA is a formal framework that appends timers to the PFA by adding the concept of Clocks to the PFA's definition and also using Clocks' timers as indicators.

In [3] it is also explained how the TPA model covers a series of classical data workflow patterns, like the parallel split pattern, the synchronization pattern, the discriminate

[5] A tool designed to elicit from the patient the amount of calories eaten in a day.

[6] $Compliant_{3w}$ is generated by another indicator that verifies that the patient has been continuously compliant for the last three weeks.

[7] For example in n_1, t_{24h} indicates a timeout occurring after twenty four hours which reactivates the nodes requiring the user to execute again all actions.

ending pattern, the milestone pattern etc. The TPA formal model is defined taking all these concepts into consideration.

By using the timed parallel automata notation, we are now formalizing the LAP conceptual model. This formalization, as we will see later in this section, ensures that the execution model is going to be finite and deterministic.

Definition 1 *A Life Assistance Protocol is a tuple* $LAP = \{A, \Delta, \Sigma, \Psi, N, Q, q_0, O, \gamma, \sigma\}$*, where:*

- R *is a finite set of typed features;*
- $A \subseteq R^*$ *is a finite set of actors, where each actor is characterized by zero or more features;*
- P *is a finite set of activities;*
- $\Delta \subseteq A \times P \times A^+$ *is the set of actions, which are activities performed by one actor and received by one or more actors;*
- C *is a finite set of Clocks;*
- T *is a finite set of time labels that can be generated by the Clock set* C*;*
- Φ *is a finite set of symbols;*
- $\Sigma \subseteq T \cup \Phi^+ \cup T \times \Phi^+ \cup \{\lambda\}$ *is the finite input alphabet;*
- $\Psi \mid \psi_i \;\; C^* \times R^* \rightarrow \Sigma$ *is the set of all possible indicators, where an indicator is a function of time (indicated by clocks) and actor's features to the input alphabet ;*
- $N \subseteq \Delta^* \times \Psi^*$ *is a finite set of nodes, where each node contains zero or more actions and zero or more indicators;*
- $Q \subseteq N^+$ *is a finite set of stages;*
- $q_0 \in Q$ *is the initial stage;*
- $O \subset Q$ *is the set of objective stages;*
- $\gamma \;\; N^+ \times \Sigma \rightarrow N^+$ *is the transition function between nodes;*
- $\sigma \;\; Q \rightarrow Q$ *is the transition function between stages.*

It is clear that there are differences between LAPs and TPAs but they can be summarized saying that:

- The LAP model explicitly defines the actors involved in the modeled workflow, along with their characteristic features;
- In LAPs we refer to stages instead of states, because in our vision they better incorporate the kind of complexities that can be found while trying to model the health condition of a person;
- The finite set of indicators used in the LAP's definition produces elements of the finite input alphabet used in the TPA's definition. In the LAP's definition we assume that the Clocks used in the TPA, and their temporized labels, may be directly used as indicators;
- The set of actions of a LAP is equivalent to the TPA's set of processes, $P_{\mathrm{TPA}} \equiv \Delta_{\mathrm{LAP}}$, but adding to the TPA's processes an executor and addressees. Thus, along with the previous point, let us conclude that $N_{\mathrm{TPA}} \equiv N_{\mathrm{LAP}}$.
- At this point we want to stress that, despite the fact that both models do not include the same concepts, the input alphabet remains regular, and the transition functions are still compatible, therefore we ensure that the execution model is going to be based on a finite and deterministic formalism.

2.3. LAP Execution Model

In addition to the workflow model specification, an execution model must also be specified. The execution model prescribes the actual implementation technology used to enact the LAP templates.

The model must support multiple LAP instances and their tasks concurrently executed, but support also autonomy, flexibility and interaction with the users. In order to support these features, we decided to use an existing workflow engine to implement the LAP Execution Model, as it provided us a quick and proved solution to implement the LAP concept and validate its efficacy. The workflow engine selected is the JBOSS JBPM [6], an Open Source workflow management system based on J2EE, running on a JBoss application server and therefore compatible with the PIPS Platform Reference Architecture.

This engine allows to represent workflows according to several standards, including an internal format, and provides tools and library to create and make persistent instances of the specified workflow as well as enacting those instances executing the tasks specified for each workflow nodes and managing the nodes transition. We have taken this component and adapted it in order to allow the specification of LAP throw its internal language and then to execute LAP instances associated to each user that is participating in a LAP. Moreover, we have created tools to simulate the LAP execution and validate them, verifying that the correct services (actions) have been invoked, the correct indicators evaluated and the appropriate transitions (reactions) enforced.

2.4. LAP Representation Language

As it was mentioned before, the LAP's definition language is be dependent on the execution model implemented. It should express the workflow process identified by the experts by using sections that allow them to write the nodes, stages, transitions and the actions executed by the nodes. The possible actions that they can select are expressed as services and the entire workflow can be understood as the execution of those services. Our model is focused on the process and the way it should be executed by carrying out actions and evaluating their results, it has also a kind of "supervisor" which are checking the decisions, the results and the possible interactions of them.

Comparing this approach, which is basically oriented to the guidance of the execution of the actions, to already existing guideline representation models, the Arden Syntax and GLIF approaches are more focused on the guidelines standardization, while PROforma on execution aspects [1], but oriented to facts declaration. Finally, Asbru defines a guideline representation language that has a very rich set of temporal constructs, however an execution engine, which is able to handle the complexity of Asbru plans, is very complicated due to the powerful features of Asbru, although there exists some execution engines based on relaxed versions of Asbru, like Asbru Light or Arbru Light+ [4]. Another difference, and the most important one is the possibility of inferring the LAP model from examples, since the LAP complexity is bounded to regular grammars, as is demonstrated in [3].

So far, the LAP templates have been described using two distinct artefacts: the first is a human readable document, organized in tables, that allows to describe the different parts of the Conceptual Model (nodes, indicators, actors, ect.) and to interact with domain experts to fill it. The second formalism is used to represent a machine readable version of the template and its instances and is based on XML, since it can be easily used to describe the workflow, the nodes and its transitions, and also to reflect the tasks and actions per node, and the stages that can be extracted from the workflow. The choice of this second formalism is dependent on the technology chosen for the Execution Model, as discussed in the previous section.

Of course, when the Framework will be completed, there will exist software tools that will allow the domain experts to represent a LAP template using the first, high-level, formalism and translate it into the second, machine-readable, formalism. For the moment, the translation work as been carried out by hand by the implementers of the LAP engine.

3. Implementation and Evaluation Plans

The proposed model is currently being used and tested in the context of the PIPS Project (Personalised Information Platform for Health and Life Services, [2]), an e-Health Integrated Project funded by the European Commission under the Sixth Programme Framework call, that aims at creating novel healthcare delivery models by building an holistic environment for Health and Knowledge Services Support.

In the PIPS project, major attention is dedicated to the issue of promoting compliance to the medical advices. The PIPS philosophy, in accordance with recent research in health promotion, is that the patient/citizen has to have control of his own behavior, in order for the advice to be completely understood and put into practice.

Such a context perfectly marries the LAP philosophy and the LAP framework will be employed in the next months to model several clinical and lifestyle related scenarios, from supporting people suffering from heart failure in their everyday chores to motivating overweight people in following a personalized nutritional plan. The first results of this experimentation should be available by the end of the current year and a thorough validation and evaluation process will be completed to understand advantages and limitations of the proposed approach.

Once the LAPs are described for the selected use cases, a twofold validation will be carried out. On the first place and before the execution model is created, a set of experts will be elicited for the validation of the LAP model and the first LAP templates. Those experts will be selected so that to have both a medical background and expertise in e-health and process management, in order that the model can be analyzed in detail in personal interviews. On the second place, after the templates are implemented in an execution solution, the instantiated LAP will be validated by controlled pilot studies, including real actors and real health care organizations, to assess its applicability and evaluate if those experiences show a tendency to the desired impact. As a result of these phases, a refinement of the templates and the model will be performed, setting the basis for the dissemination of them in wider forums.

4. Conclusions

In this paper we introduced the Life Assistant Protocol model, a concept coming from the classical medical guidelines but, now, framed in the new p-Health paradigm. This approach has been presented in a framework, including the Conceptual Model and the Workflow Model, which are the theoretical base of the LAP, and the Execution Model, that will allow us to put in practice the LAP concept. The PIPS Project represents our development environment, in which we will be able to validate and evaluate the complete framework, also receiving feedback from real users in real situations. The project will be deployed in San Raffaele Hospital (Milan, Italy) in the next months and in La Fe Hospital (Valencia, Spain) at the end of the project, giving us the opportunity to evaluate the effectiveness of our approach.

Acknowledgements

We would like to thank the European Commission for funding the PIPS Project (Contract 507019), where all this work is being carried out, as well as to the other partners who are helping in the project development.

References

[1] P.A. De Clercq, J.A. Blom, H.H.M. Korsten, and A. Hasman, 'Approaches for creating computer-interpretable guidelines that facilitate decision support', *Artificial Intelligence in Medicine*, **31**(1), 1–27, (2004).

[2] PIPS Consortium. PIPS - Personalised Information Platforms for Life and Health Services. http://www.pips.eu.org, 2004.

[3] C. Fernández and J.M. Benedí, 'Activities based on Workflow Mining: A Theoretical framework', in *Proceedings of I Workshop on Technologies for HealthCare and Healthy Lifestyle, WTHS'06*, (April 2006).

[4] C. Fuchsberger and S. Miksch, 'AsbruÕs Execution Engine: Utilizing Guidelines for Artificial Ventilation of Newborn Infants', in *Proceedings of the joined Workshop Intelligent Data Analysis in Medicine And Pharmacology and Knowledge-Based Information Management In Anaesthesia and Intensive Care (IDAMAP and KBIM-AIC 2003), in conjunction with the 9th Conference on Artificial Intelligence in Medicine in Europe (AIME 2003)*, (2003).

[5] C.G. Hagerty, D. Pickens, C. Kulikowski, and F. Sonnenberg, 'HGML: a hypertext guideline markup language', in *Proceedings AMIA Symposium*, pp. 325–329, (2000).

[6] JBoss. JBPM Home Page. http://www.jboss.com/products/jbpm, 2007.

[7] OpenClinical.org. Methods and tools to support the computerisation of clinical practice guidelines: a short introduction. http://www.openclinical.org/gmmintro.html, 2005.

[8] OpenClinical.org. Clinical pathways: multidisciplinary plans of best clinical practice. http://www.openclinical.org/clinicalpathways.html, 2006.

[9] M Peleg and et al., 'Guideline Interchange Format 3.5 Technical Specification', *Intermed Collaboratory*, (May 2004).

[10] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. Greenes, N. Johnson, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, and M. Shortliffe, 'Comparing Computer Interpretable Guideline Models: A Case-Study Approach', *JAm Med Inf Assoc*, **10**(1), 52–68, (Jan-Feb 2003).

[11] Yuval Shahar. Automated Support to Clinical Guidelines and Care Plans: The Intention-Oriented View. Open Clinical Briefing Papers, http://www.openclinical.org/briefingpaperShahar.html, 2002.

[12] R.N. Shiffman, G. Michel, A. Essaihi, and E. Thornquist, 'Bridging the guideline implementation gap: a systematic, document-centered approach to guideline implementation', *J Am Med Inform Assoc*, **11**(5), 436–438, (Sept.-Oct. 2004).

[13] AD. Sutton and J. Fox, 'The Syntax and Semantics of the PROforma Guideline Modeling Language', *J Am Med Inform Assoc.*, **10**(5), 433–443, (2004).

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-203

DeGeL: A Clinical-Guidelines Library and Automated Guideline-Support Tools

Avner HATSEK[1], Ohad YOUNG, Erez SHALOM and Yuval SHAHAR
Medical Informatics Research Center, Department of Information Systems Engineering, Ben Gurion University, Beer-Sheva, Israel

Abstract. Using machine-interpretable clinical guidelines to support evidence-based medicine promotes the quality of medical care. In this chapter, we present the Digital Electronic Guidelines Library (DeGeL), a comprehensive framework, including a Web-based guideline repository and a suite of tools, to support the use of automated guidelines for medical care, research, and quality assessment. Recently, we have developed a new version (DeGeL.NET) of the digital library and of its different tools. We intend to focus in our exposition on DeGeL's major tools, in particular for guideline specification in a Web-based and standalone fashion (Uruz and Gesher), tools for search and retrieval (Vaidurya and DeGeLookFor) and for runtime application (Spock); and to explain how these tools are combined within the typical lifecycle of a clinical guideline.

Keywords. Clinical Guidelines, Digital Libraries, Medical Decision-Support Systems, Knowledge Representation.

Introduction

Over the past 20 years, there were multiple efforts to provide automated support to evidence-based medicine by formalizing clinical **guidelines** (**GLs**) into machine interpretable formats [1,2]. GLs represented in a machine-comprehensible, formal format, can be applied by a computerized agent as a tool to support physician decisions at the point of care, or as a tool for retrospective quality assessment and research. Several guideline-specification ontologies such as GLIF, GEM, and Asbru were developed to represent guidelines in a formal and machine interpretable format. In our research, we focus on the Asbru ontology as our target specification ontology. In this chapter we present a brief up to-date exposition of a Digital Library for GLs that we had developed. We explain how it assists physicians and knowledge engineers in acquisition and representation of the GL's knowledge and how it assists users in searching, retrieving, and applying a relevant GL in a manner customized to each patient.

[1] Corresponding Author address: Dpt. of Information System Engineering, Medical Informatics Research Center, Department of Information System Engineering, Ben Gurion University of the Negev, P/O/B/ 653, Beer Sheva, 84105, Israel; E-mail: hatsek@bgu.ac.il.

1. The DeGeL Framework and its Hybrid Meta Ontology

In order to support the automation of guideline-based care, there is a need to convert the GL from its free-text representation into a machine interpretable format. The guiding principle followed in our research is specifying GLs through the collaboration between expert physicians, clinical editors,(i.e. physicians who are not necessarily experts in the GL medical domain, but are familiar with the semantics of the representation format, known also as the GL ontology), and knowledge engineers. Expert physicians and clinical editors transform the clinical knowledge represented in free text GLs into intermediate, semantically meaningful representations while knowledge engineers convert these intermediate representations into a formal, machine-interpretable representation.

To convert clinical GLs into a formal representation, we have developed the **Digital electronic Guidelines Library** (**DeGeL**) [3], which uses a *hybrid* representation, namely a representation methodology that includes several intermediate formats; these formats are increasingly formal. All intermediate and final formats are stored within the knowledge base. The current representation formats include (1) the original *full text*, (2) a *semi-structured* representation (marked-up text), (3) a *semi-formal* representation which includes control structures such sequential or parallel sub-plans order, and (4) a *fully formal*, machine-comprehensible format. The intermediate representation levels have additional benefits; the semi-structured level is crucial for context-sensitive search of GLs; the semi-formal level supports application of the GL by a clinician at the point of care, without access to an **electronic medical record** (**EMR**).

The DeGeL library uses multiple GL ontologies by which GLs can be represented. Each of these ontologies consists of knowledge-roles that are semantic fields within the ontology, such as “Eligibility Conditions”. DeGeL uses a Hybrid *Meta-Ontology* that is common to all specific GL ontologies. The Hybrid Meta-Ontology is composed from two major components; documentation *ontology* and a *specification meta-ontology*:

- The documentation ontology includes documentary knowledge roles that are common to all guideline ontologies. GL’s title, authors and *semantic classification axes* that index GLs by several conceptual indices (e.g., disorders, therapies) are examples of common elements. The documentation ontology distinguishes between *source* GLs and *hybrid* GLs and provides different documentation elements for each of these GL types. *Source* GLs are free-text (html) documents uploaded and stored in DeGeL, *Hybrid* GLs (also known as "mark-ups") are the products of the (mark-up) specification process.
- The specification meta-ontology includes elements common to all guideline ontologies, for example the hierarchical structure of the target-ontology's knowledge roles. The specification meta-ontology is being used to define multiple target specification ontologies (e.g. GLIF, Asbru, GEM) that will be used for GL representation.

2. DeGeL.Net

To provide several improvements to the web-based architecture of the previous version of DeGeL, we have developed a new version (**DeGeL.NET**) of the digital library and of its different tools. Our goal when designing the new version was to create a distributed, web-service based, open architecture implementation according to the *Service Oriented Architecture* (*SOA*) design specification [4]. This new approach grants the ability to develop a suit of tools for guideline specification, retrieval and application. In addition, the open architecture may host alternative tools for guideline specification and application. The DeGeL.NET implementation includes the following main modules: (1) a knowledge base server, (2) a guideline-specification tool (*Gesher*), (3) a runtime application engine for clinical guidelines (*Spock*).

DeGeL's server allows development of rich client tools by using web-service methods to retrieve and edit guidelines in the knowledge-base. The server's architecture is assembled of the following modules: (1) a guideline database that contains the overall schema to support the hybrid multiple ontology representation, (2) a module that is responsible for guideline-knowledge creation, reading, updating, and deletion, (3) a new guideline search engine, named **DeGeLookFor**, which is intended to replace the current search engine, **Vaidurya** [5], which facilitates full-text, context-sensitive and concept-based search methods for enhanced guidelines retrieval[6], (4) an authorization & authentication module, which supports the group-based authorization model, and (5) a web-service API that enables the guideline knowledge-base server to accept client requests and to orchestrates multiple steps, in order to perform the requested transactions.

We have developed additional client tools to allow DeGeL's administrators to perform tasks to maintain the guideline library, such as management of ontologies or granting user permissions:

- **OntologyBuilder** supports the task of acquisition and maintenance of the hybrid ontologies stored within the knowledge-base of DeGeL server. The implementation of DeGeL's hybrid Meta-Ontology enables library administrators to create and modify the structure of ontologies and to define the type of content for each knowledge role of the acquired ontologies. guideline knowledge representation.
- **DeGeLock** is a client tool developed for the administration of *DeGeL's* authorization module. The library administrators *use DeGeLock* to create and manage groups of users with deferent profiles consisting of a set of library roles, which collaborate in acquiring new guidelines into the knowledge base.
- **AxisBuilder** was developed to create and maintain the semantic axes of medical concepts used to classify (index) guidelines, in order to support enhanced concept-based retrieval by DeGeL's search engine (DeGeLookFor).

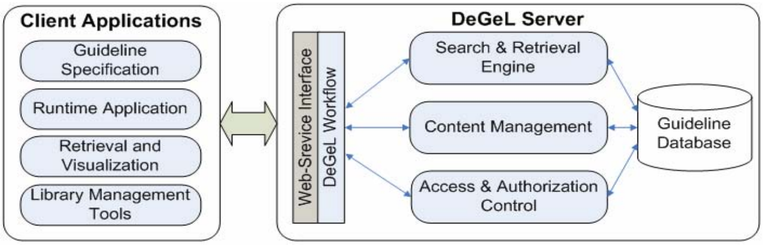


Figure 1. The general architecture of DeGeL.Net framework. The server consists of GL database, content management module, retrieval engine, and access and authorization control module. The client application uses a web-service based API to connect to the server and perform task of GL specification, retrieval, visualization and application.

3. Gesher – A Graphical Framework for Specification of Clinical Guidelines

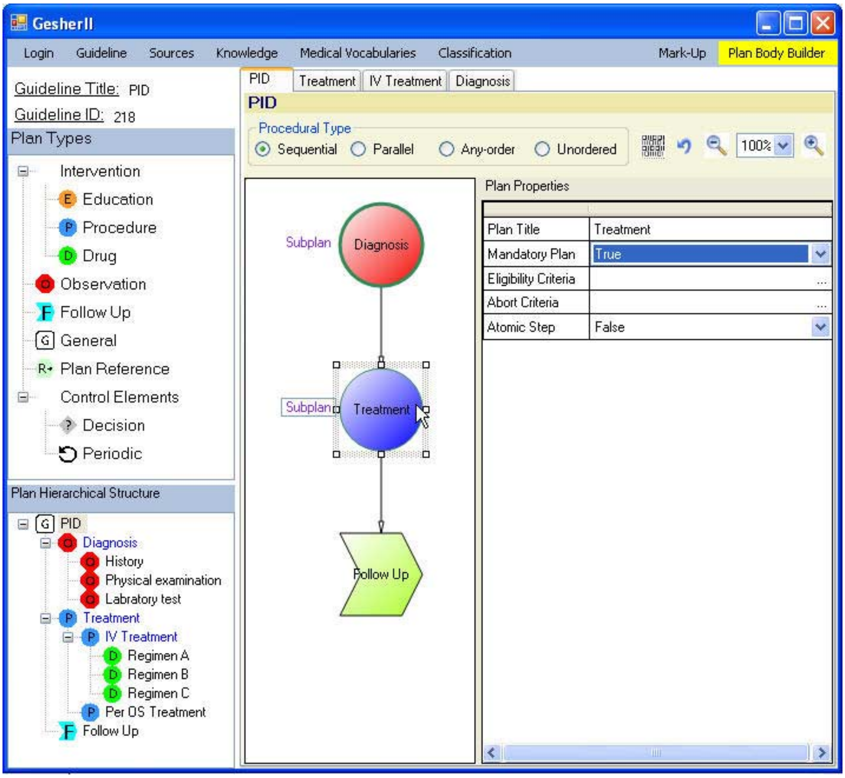
The Gesher system is a client application designed to support the process of incremental guideline specification at multiple representation levels according to target specification ontology. It is intended to supplement and eventually replace the Uruz Web-based markup tool [3] whose feasibility for use by expert-physician editors was demonstrated in an extensive evaluation [7]. Gesher is designed to support the gradual knowledge acquisition process performed by a collaboration of an expert physician familiar with the domain-specific clinical knowledge, a clinical editor familiar with general medical knowledge and with the semantics of the target GL ontology, and a knowledge engineer. This collaboration, which includes specification of the guideline ontology specific consensus, is critical for achieving high quality specification. Gesher supports the gradual process through all the steps of creating formal representation of the guideline and for further maintenance and modification of the guidelines. A methodology for this specification process was developed and evaluated, using the URUZ tool. The methodology consists of the following major steps:

1. Creation of a clinical consensus – Expert physicians deciding on the recommendation to be adapted from the source guidelines; the adaptation may include changes to fit the local clinical setting.
2. Creation of an Ontology Specific Consensus (OSC) – expert physicians and knowledge engineers collaborate to specify the knowledge of the guideline according the selected ontology.
3. Mark-up: The clinical editor is creating a semi-structured representation of the guideline, according the OSC.
4. Specification of the semi-formal and formal representation by the knowledge engineer. Currently the semi-formal and formal specification uses Asbru ontology [8]. Asbru is a GL representation language that enables GL specification using a hierarchical (plan, sub-plan) representation and an expressive temporal specification language for Actions (e.g. periodic plans), Data Abstractions (e.g. clinical temporal patterns), Conditions (i.e. conditions to enter or abort or complete a plan) and intentions (e.g. process, outcome).

In order to support the steps of the specification methodology Gesher must provide the ability to perform the following tasks:

3.1. The Decomposition of the Guideline

As part of the consensus formation process, Gesher supports the decomposition of the GL into plans and sub-plans and creation of the GL's procedural control flow. The Control Flow is one of the important aspects of the GL, which represents the evidence-based clinical algorithm that is recommended by the guideline authors. Most of the approaches for GL specification contain a hierarchical structure for representing the overall complex GL. Gesher supports the creation of this hierarchical structure and provides the ability to explore the hierarchy and the control flow in deferent views such as tree-structure and control flow diagrams (Figure 2). The hierarchical structure of the GL is stored in DeGeL along with elements of procedural and declarative knowledge that are stored for each sub-plan. This first product of specification is later being translated into knowledge roles of the target ontology, such as Asbru's "plan-body" or "complete-condition". By supporting tools to apply these steps of the specification methodology, Gesher supports the specification into multiple target ontologies according to the Hybrid Meta-Ontology.



Guideline Title: PID

Guideline ID: 218

PID

Procedural Type

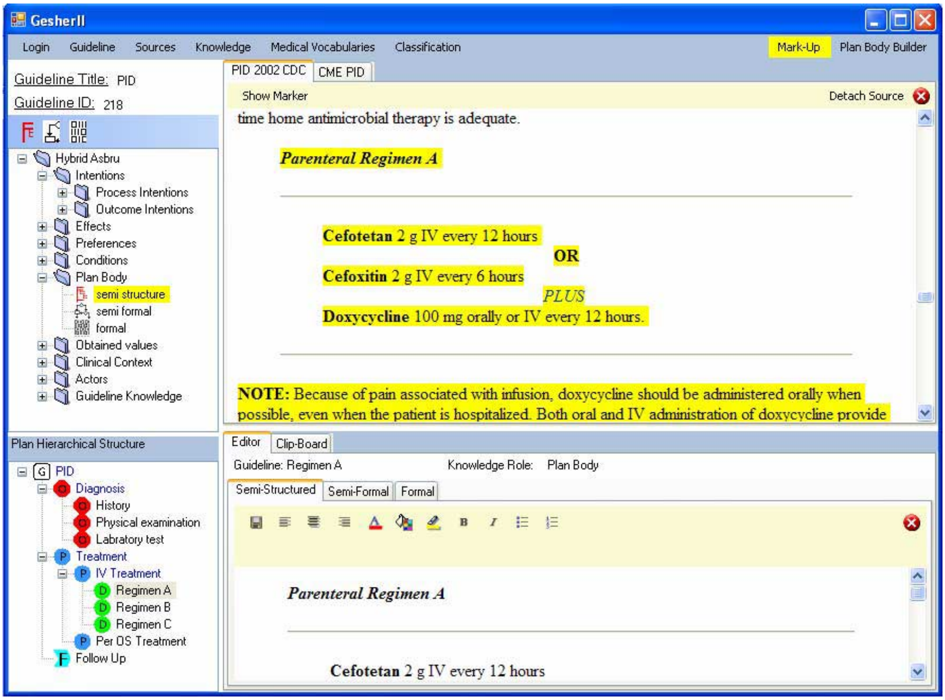
● Sequential ○ Parallel ○ Any-order ○ Unordered

Plan Title	Treatment
Mandatory Plan	True
Eligibility Criteria	...
Abort Criteria	...
Atomic Step	False

Figure 2. The Hierarchical Plan Builder in Gesher used by the expert physicians for specifying the procedural aspects of the guideline. In this case the root plan of the GL for treating pelvic inflammatory disease composed of three different sub-plans that should be performed in sequence

3.2. Specification of the Semi-Structured Representation

The next step to follow after the OSC is created is to create the Semi-Structured representation level (a mark-up, structuring of free text process). This representation level is usually created by the clinical editor. Each plan and sub-plan is constructed from elements according to the knowledge roles of the selected target ontology (e.g. Abort-Condition or Evidence-Level). In order to create a Semi-Structured representation which is an intermediate one but not yet a formal one, a clinical editor needs to refine the plans, created in the Ontology Specific Consensus, and link those plans to portions of text from the original text-based GL, where the evidence-based recommendations are obtained. To edit the semi-structured representation content (Figure 3), Gesher provides a rich HTML editor, to help the user to perform mark-up by dragging portions of labeled content from one or more source guidelines into the selected knowledge roles frames.



Guideline Title: PID

Guideline ID: 218

time home antimicrobial therapy is adequate.

Parenteral Regimen A

Cefotetan 2 g IV every 12 hours

OR

Cefoxitin 2 g IV every 6 hours

PLUS

Doxycycline 100 mg orally or IV every 12 hours.

NOTE: Because of pain associated with infusion, doxycycline should be administered orally when possible, even when the patient is hospitalized. Both oral and IV administration of doxycycline provide

Guideline: Regimen A

Knowledge Role: Plan Body

Parenteral Regimen A

Cefotetan 2 g IV every 12 hours

Figure 3. The Interface used to Edit Semi-Structured Representation Level. The clinical editor creates the "Regimen A" sub-plan of the PID IV treatment by marking the text from the source GL

3.3. Specification of the Semi-Formal representation

The Semi-Formal representation level is described using graphical widgets such as the Expression-Builder (Figure 4) which is used by the clinical editor or knowledge engineer to create logical expressions from several types such as "And-Or trees" or "Switch-Case" expressions. The Semi-Formal representation level is used by the runtime application engines for applying plans using the system's user (e.g. physician or nurse) as a mediator to the patient data. Semi-Formal representation defers from the Semi-Structured mainly by a more formal description of expressions within the GL. Our motivation for developing intuitive graphic controls is to allow the clinical editors to be involved at semi-formal specification.

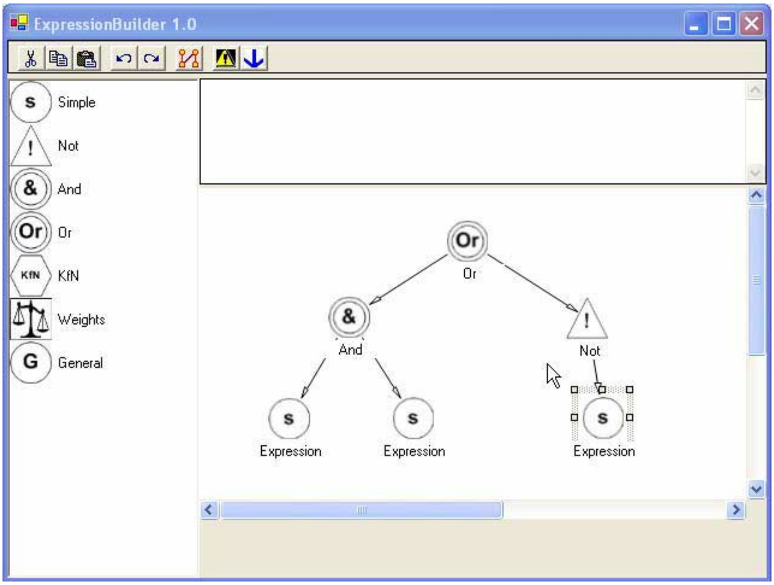


Figure 4. The Expression-Builder which is used by the clinical editor or knowledge engineer to create Semi-Formal expressions

3.4. Specification of the Formal representation

The Formal representation of a GL is obtained in order to execute the GL, by a computer agent, to perform tasks such as decision support or retrospective quality assessment; therefore, it should include a formal definition for each aspect of the GL. While dealing with the control flow and logical expression in the intermediate representation levels, in the Formal level it is required to create a complete definition for executing the expressions within the GL for a given patient. These expressions include requirements for patient data, and in many GL scenarios, will consist of temporal aspects. The knowledge described within these expressions is of a declarative nature. The overall architecture for the application of formal GLs includes different modules that are being involved. One important module is for mediating between the GL application engine and the patient related data. In the architecture we had developed, this mediator is used not only for accessing the raw patient data, but also to provide knowledge-based temporal reasoning about these data [9]. The Mediator module includes a knowledge base to store the relevant information needed for providing temporal reasoning, and also a temporal declarative knowledge acquisition tools. Gesher integrates with the temporal knowledge acquisition tool and with its knowledge base in order to enable full and formal specification of the GL, including the data and temporal aspects described in the clinical algorithm.

When considering the clinical knowledge embedded in a clinical guideline, we define two types of knowledge: The first is the procedural knowledge representing the steps of the protocol and is described using the specification ontologies. The second type is declarative knowledge, which consists of the concepts in the protocol and their definition according to the context in which they are relevant (e.g. in the context of Hypothyrd guideline, Hypothyroidism is defined as TSH ≤ 0.4 IU/ml and FT3>4.2pg/ml). To accomplish the task of representing declarative knowledge at formal level, Gesher uses the interface of temporal abstraction knowledge acquisition tool, which is one of the tools used in the IDAN architecture.

Finally, Gesher has access to the MEIDA system [10], a comprehensive framework for linking medical decision-support systems to clinical databases, which includes a search engine for controlled medical vocabularies such as ICD-9, CPT and LOINC. Gesher uses the MEIDA system to enable the use of standards terms from controlled medical vocabularies when defining the clinical terms of the guideline, thus making it sharable and reusable across multiple local clinical-database platforms.

3.5. GL Classification

DeGeL library contains structures of semantic indices for GL classification (of both original sources and formal mark-ups). These indices are used to gain better retrieval abilities when users search for knowledge in the library. The clinical editors can use Gesher to classify the new GLs that are added into the library using the IndexieGuide graphical widget.

4. Spock – Runtime Application of Hybrid-Asbru Clinical Guidelines

DeGeL support runtime application of intermediate-represented Hybrid-Asbru GLs, with or without an available EMR, capitalizing on the DeGeL framework as a knowledge repository of machine-interpretable guidelines and on the IDAN architecture for access and sophisticated querying of heterogeneous clinical-data repositories. The new approach was implemented as the Spock system [11], which provides the necessary functionality to support the task of applying clinical guidelines at the point of care.

The Spock system's architecture includes the Spock engine responsible for the actual interpretation of the knowledge encoded in the intermediate-represented guidelines, and a Spock server, which provides remote services to store and retrieve the history of guideline applications from a GL application log repository, and remote external services, such as the DeGeL server’s services for retrieving GL's knowledge.

4.1. The Hybrid Runtime Application Model of the Spock System

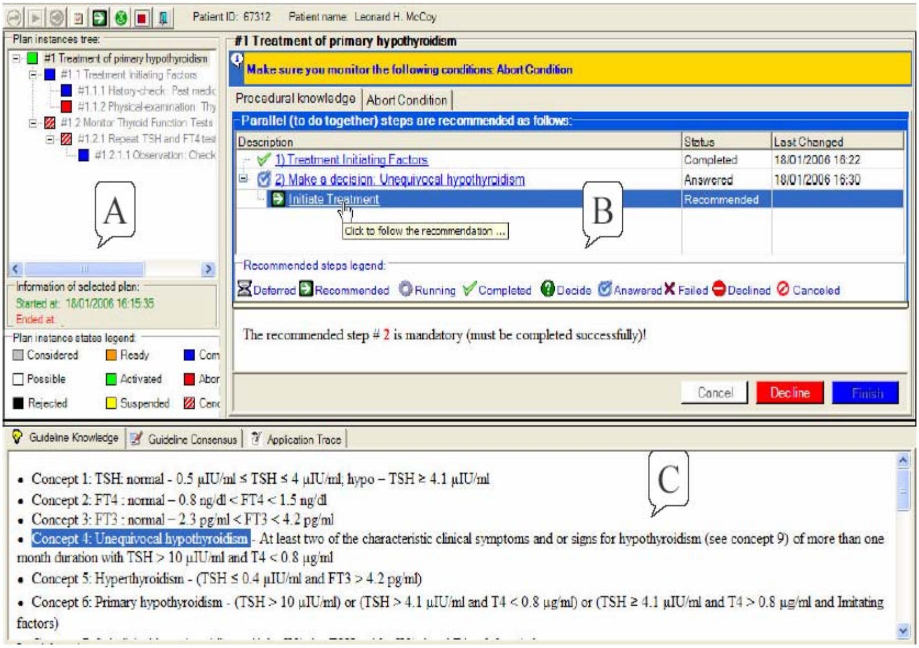
The hybrid runtime application model of Spock allows the application of GLs in several scenarios implied by the level of representation of the currently applied GL, and by the availability of an effective electronically patient data. Each GL can be represented in more than one representation level, for example, a fully formal representation of the GL’s abort-condition, and semi-structured text or semi-formal representations for the GL’s plan-body.

Elements that are represented only in semi-structured text can be displayed to the care provider during the application process to reduce the workload by providing direct access to the content which is relevant in each specific step of the overall GL. Elements which are represented in the semi-formal level are used by Spock to guide more closely the care-provider through the stages of the GL application.

The semi-formal Asbru is expressive enough to fully describe the control structure of the GL’s plan-body (e.g. sequential, concurrent or periodic combination of sub-plans), thus, allowing Spock to assist the care-provider in applying the GL from beginning to end. The semi-formal Asbru expressions describe clinical decisions within the GL. These decisions are usually evaluated according to the patient data. In the case where EMR is not available, the expressions are being presented to the care-provider for evaluation, and the results of the evaluation will be used by Spock to determine the next appropriate step according to the GL. A fully automated application of GL is feasible when the patient EMR is available and the GL is represented in formal level. In this scenario Spock will use the mediator to the patient data for evaluating the GL expressions, and will provide recommendations and send alerts to the care-provider according the GL control structure.

4.2. The Guideline Application Log

Storing the GL application history is necessary to support the time spanning nature of the application process wich is usually performed in an episodic fashion during patient's visit. The GL application log consists of detailed records for each session of the GL application. One major purpose of the application log is to allow resuming GL application which was previously paused. In order to enable GL application in a distributed environment, the GL application Log is stored in a repository which is located on a remote server which is available to all instances of the Spock client application. The application log may be used for other purposes, such as retrospective quality assessment of the medical care.


Patient ID: 67312
Patient name: Leonard H. McCoy
Plan instances tree:
#1 Treatment of primary hypothyroidism
#1.1 Treatment Initiating Factors
#1.1.1 History-check: Past medic
#1.1.2 Physical-examination: Thy
#1.2 Monitor Thyroid Function Tests
#1.2.1 Repeat TSH and FT4 test
#1.2.1.1 Observation: Check
A
Information of selected plan:
Started at: 18/01/2006 16:15:35
Ended at:
Plan instance states legend:
Considered
Ready
Com
Possible
Activated
Abor
Rejected
Suspended
Canc
#1 Treatment of primary hypothyroidism
Procedural knowledge
Abort Condition
B
Click to follow the recommendation ...
Recommended steps legend:
Deferred
Recommended
Running
Completed
Decide
Answered
Failed
Declined
Canceled
Cancel
Decline
Finish
Guideline Knowledge
Guideline Consensus
Application Trace
C

Make sure you monitor the following conditions: Abort Condition

Parallel (to do together) steps are recommended as follows:

Description	Status	Last Changed
✓ 1) Treatment Initiating Factors	Completed	18/01/2006 16:22
☑ 2) Make a decision: Unequivocal hypothyroidism	Answered	18/01/2006 16:30
Initiate Treatment	Recommended	

The recommended step # 2 is mandatory (must be completed successfully)!

- Concept 1: TSH: normal - 0.5 μIU/ml ≤ TSH ≤ 4 μIU/ml; hypo – TSH ≥ 4.1 μIU/ml
- Concept 2: FT4 : normal – 0.8 ng/dl < FT4 < 1.5 ng/dl
- Concept 3: FT3 : normal – 2.3 pg/ml < FT3 < 4.2 pg/ml
- Concept 4: Unequivocal hypothyroidism - At least two of the characteristic clinical symptoms and or signs for hypothyroidism (see concept 9) of more than one month duration with TSH > 10 μIU/ml and T4 < 0.8 μg/ml
- Concept 5: Hyperthyroidism - (TSH ≤ 0.4 μIU/ml and FT3 > 4.2 pg/ml)
- Concept 6: Primary hypothyroidism - (TSH > 10 μIU/ml) or (TSH > 4.1 μIU/ml and T4 < 0.8 μg/ml) or (TSH ≥ 4.1 μIU/ml and T4 > 0.8 μg/ml and Imitating factors)

Figure 5. The Spock run-time guideline (GL) application module's default interface: The left panel (A) displays the plan instances that were created during GL application as a hierarchy; the top center panel (B) displays the relevant knowledge roles of the current selected plan-instance, each knowledge role in separate tab (e.g. Procedural Knowledge, Abort Conditions); The bottom panel (C) displays guideline information such as definitions of concepts used in the guideline, a detailed outline of the guideline consensus, and the progress of the application using the tabs Guideline Knowledge, Guideline Consensus and Application Trace in that order.

5. Summary

In this chapter we have presented the DeGeL framework, which we have developed in order to support the specification, storage, search, retrieval, and application of automated clinical guidelines for medical care, research, and quality assessment. In this exposition, we have focused on the major components of the framework; The GL library server, the GL specification module, and the GL runtime application. We have also presented our hybrid model for representing GL knowledge at several co-existent levels of representation, and the methodology that we propose, as well as its main participants, for the GL specification process. In future research, we intend to further improve the integration of the procedural and declarative elements of the GL knowledge, an integration that is crucial for obtaining high quality formal representation and thus for GL application and sharing. We also intend to continue evaluating the framework and its components with the collaboration of physicians in several medical domains.

Acknowledgements

This research was supported in part by NIH award LM-06806. Dr M. Goldstein, Dr S. Martins, Dr L. Basso, Dr H. Kaizer, Prof E. Lunenfeld, Dr A. Yarkoni, Dr G. Bar, and Dr A. Sobol were extremely helpful in the evaluation of all DeGeL tools.

References

[1] Peleg, M., Tu, S. W. *et al.* (2002): Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *Journal of American Medical Informatics Association* 10(1): 52–68.

[2] De Clercq, P., Blom, J. *et al.* (2004): Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial Intelligence in Medicine* 31(1): 1-27.

[3] Shahar, Y. and Young, O. (2004): A Framework for a Distributed, Hybrid, Multiple-Ontology Clinical-Guideline Library and Automated Guideline-Support Tools. *Journal of Biomedical Informatics* 37(5): 325-344.

[4] Michael, N. and Munindar, P. (2005): Service-Oriented Computing: Key Concepts and Principles. *IEEE Internet Computing* 09(1): 75-81.

[5] Moskovitch, R., Hessing, A. and Shahar, Y. (2004): Vaidurya – a concept-based, context-sensitive search engine for clinical guidelines. *Medinfo 2004* 11(Pt 1):140-4.

[6] Moskovitch, R. *et al.* (2007): Comparative Evaluation of Full-text, Concept-based, and Context-sensitive Search. *Journal of American Medical Informatics Association* 14(2): 164-174

[7] Shalom E. (2006): An Evaluation of methodology for Specification of Clinical Guidelines at Multiple Representation Levels. M.Sc. Thesis, Dep. Of Information Systems Engineering, Ben Gurion University, Beer Sheva, Israel.

[8] Miksch, S., Shahar, Y. *et al* (1997): Asbru: A Task-Specific, Intention-Based, and Time-Oriented Language for Representing Skeletal Plans. *Proceeding of the 7th Workshop on Knowledge engineering: Methods & Languages:* The Open University, Milton Keynes, UK.

[9] Boaz, D. and Shahar, Y. (2005): A Framework for Distributed Mediation of Temporal-Abstraction Queries to Clinical Databases. *Artificial Intelligence in Medicine* 34(1):3-24.

[10] German E. (2006): Linking Medical Decision Support Systems to Clinical Databases. Master Thesis, Dep. Of Information Systems Engineering, Ben Gurion University, Beer Sheva, Israel.

[11] Young, O. and Shahar, Y. *et al* (In Press): Runtime Application of Hybrid-Asbru Clinical Guidelines. *Journal of Biomedical Informatics*, In Press.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-213

A Constraint-based Approach to Medical Guidelines and Protocols

Arjen HOMMERSOM [a,1], Perry GROOT [a], Peter LUCAS [a], Mar MARCOS [b], and Begoña MARTÍNEZ-SALVADOR [b]

[a] *Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen the Netherlands*
[b] *Universitat Jaume I, Castellon, Spain*

Abstract. Medical guidelines and protocols are documents aimed at improving the quality of medical care by offering support in medical decision making in the form of management recommendations based on scientific evidence. Whereas medical guidelines are intended for nation-wide use, and thus omit medical management details that may differ among hospitals, medical protocols are aimed at local use, e.g., within hospitals, and, therefore, include more detailed information. Although a medical guideline and an associated protocol concerning the management of a particular disorder are related to each other, one question is to what extent they are different. Formal methods are applied to shed light on this issue. A Dutch medical guideline regarding the treatment of breast cancer, and a Dutch protocol based on it, are taken as an example.

Keywords. Adaptation, Verification, Model Checking

Introduction

Medical management is increasingly based on recommendations from the medical scientific community, summarised in medical guidelines and protocols. Medical guidelines are systematically developed, structured documents, containing conclusions and recommendations, based on scientific evidence [6]. These documents are called *evidence-based guidelines*. Medical protocols are local adaptations of medical guidelines.

The goal of the work described here is to better understand the differences and similarities between guidelines and protocols. First, insight is obtained into the relation of a medical guideline and protocol concerning the medical management of one particular disorder, breast cancer. Based on the results of this analysis, we have carried out a formal analysis of parts of both the guideline and the protocol for breast cancer treatment, which provides a rigorous method for finding such differences. This is done by looking at both medical protocols and guidelines as defining (logical) constraints on the medical management of patients performed in practice. This approach was inspired by a statement by Wiersma and Burgers that “recommendations in guidelines should not only be based on evidence extracted from scientific literature, but take into account the context of daily medical practice as well” [17]. In effect, this makes the comparison between guidelines

[1] Corresponding Author: Arjen Hommersom, Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, the Netherlands; E-mail: arjenh@cs.ru.nl.

and protocols more realistic. In principle this approach would allow one to discover flaws or suboptimal management actions in the medical management in practice, assuming that a given protocol and guideline are correct, or to find incorrect or suboptimal medical management decisions in a protocol or guideline, assuming that the medical management in practice is correct and optimal. In the research described in this chapter we investigate whether this is possible using a combination of informal and formal, in particular model checking, methods.

1. Medical Guidelines and Protocols

A medical guideline is an extensive document, developed by a working group involving professionals involved in the management of the disorder covered by the guideline. By definition, a protocol is seen as a local version of a guideline, meant to be useful as a guide for daily clinical care. Hence, basically, a medical protocol is a summary of the most important sections that are in the guideline, mostly recommendations, supplemented with hospital-specific details concerning the treatment. This implies that many sections in a protocol may be very similar to related sections in a guideline. However, there may also be differences, partly due to differences in opinion between the guideline designers and protocol designers, and partly due to the difference in purpose of a guideline and protocol. The guideline that we have used in this study is the 2004 version of the Dutch CBO guideline on the treatment of breast cancer. The protocol that we have used is the protocol of the Dutch Integral Cancer Centre East (IKO in Dutch), which is based on the CBO guideline. To understand the differences between breast cancer treatment in the guideline and the protocol, we briefly review the result of an informal analysis of cases where the recommendations of the guideline and protocol differs.

Most of the differences between the CBO guideline and the IKO protocol that were found are due to the fact that the protocol is more specific than the guideline. Such differences are also referred to as ‘cookbook’ difference, indicating that the difference implies an insignificant refinement compared to the guideline. For example, in the protocol, an ultrasound axilla is suggested as default action during the sentinel node procedure to assess the stage of the disease, whereas the guideline does not provide such a default. Similarly, a protocol may choose a particular order between interventions to improve the efficiency of the health care process, whereas a guideline does not recommend any.

There are few real differences (i.e., differences that cannot be described in terms of a refinement) between the protocol and the guideline. The main reason for this is that this particular protocol is heavily based on the guideline and the developers are involved in both the guideline and the protocol. As a consequence, the evidence used is the same in both cases, and therefore the recommendations are very similar. An example of such a real difference is the case when tumour cells are isolated in the sentinel node. According to the guideline, axillary treatment can be omitted, in contrast to the IKO protocol where additional axillary treatment is recommended in any case. This difference is a significant change in advise and, assuming a closed world assumption on the interventions that may be performed, can be seen as a contradiction. However, it may be argued that the evidence underlying the guideline advice was based on retrospective studies and is therefore uncertain.

2. Medical Management in Breast Cancer

First, we give an informal description on the medical management as stated in the CBO guideline (and IKO protocol) that deals with locoregional treatment of operable breast cancer, i.e., T1-2 N0-1 M0 breast cancer according to the TNM classification system [8]. Thereafter, we discuss temporal logic as a means for formalising the medical management of breast cancer.

2.1. Informal Description of Medical Management

According to the CBO guideline there are only two options for local treatment of operable invasive breast cancer: breast-conserving therapy (BCT) or modified radical mastectomy (MRM). BCT implies ample local excision of the tumour, an axillary staging procedure, and radiotherapy of the breast. MRM involves a total resection of the breast (mastectomy) and dissection of the axillary nodes (AND). The aim of BCT is to achieve a survival rate comparable to MRM with an optimal cosmetic result in terms of the treated breast. BCT is usually the preferred treatment unless the patient has a clear preference for MRM and there are no contra indications for BCT, i.e., there is no (1) multicentricity (two or more tumour foci in different quadrants), or (2) diffuse malignant microcalcifications, or (3) previous radiotherapy of the breast. Another contra indication for BCT is obtained *during* surgery: (4) the margins of the local excision remain tumour-positive after repeated local excision attempts. In this case, local excision attempts are unsuccessful in removing the primary tumour and treatment therefore switches to MRM.

Treatment of the axillary nodes is also part of the treatment of breast cancer as the pathologic assessment of axillary lymph nodes remains the most important prognostic variable for the invasive breast cancer patient. An optimal assessment would be achievable by means of a complete axillary node dissection (AND). However, AND may lead to morbidity, e.g., pain, limited shoulder movement. An alternative for axillary staging is the sentinel node procedure (SNP), which only dissects the sentinel nodes, i.e., those nodes that drain the area of the breast where the primary tumour is located and thus are most likely to contain metastasis. The SNP is currently the standard procedure for axillary staging in breast cancer provided that there are no contra-indications, where contra-indications of SNP are defined as (1) suspected/proven malignancy in the axillary nodes, (2) tumour $>$ T2, (3) multiple tumour foci, and (4) potentially disrupted lymph drainage due to recent axillary surgery or a large biopsy cavity following tumour excision.

When the SNP is not possible, complete axillary node dissection should be carried out. Furthermore, treatment of the axilla is indicated (i.e., dissection, radiotherapy) for all forms of lymph node metastasis.[2]

2.2. Temporal Logic Representation

The CBO guideline can be interpreted as (temporal) constraints on medical management. It has been shown in [10] that the step-wise, possibly iterative, execution of a guideline can be described precisely by means of temporal logic. The logic that we use here for specifying properties of medical guidelines is a combination of Computation Tree Logic (CTL) [3,4] and Linear Temporal Logic (LTL) [13].

[2] Here, the CBO guideline differs from the IKO protocol as it makes an exception for isolated tumour cells.

CTL uses atomic propositions and Boolean connectives (e.g., $\neg, \vee, \wedge$) to build up more complicated expressions for describing properties of states. Furthermore, CTL formulas can be composed of *path quantifiers* and *temporal operators* for describing properties of *computation trees*, i.e., all paths that are possible from a certain state. The path quantifiers **A** and **E** specify that all of the paths or some of the paths starting at a specific state have some property. The temporal operators describe properties of a path through the tree. The four temporal operators are **X**, **G**, **F**, and **U**, where $\mathbf{X}\varphi$ is true if φ holds in the next state, $\mathbf{G}\varphi$ if φ holds in the current and in all future states, $\mathbf{F}\varphi$ if φ holds in some state in the future (or is true in the current state), and $\varphi\mathbf{U}\psi$ if φ holds until ψ holds.

LTL provides operators for describing events along a single computation path. Each formula is of the form $\mathbf{A}f$, where f is a path formula, which is either an atomic proposition or inductively defined as $\neg f$, $f \vee g$, $f \wedge g$, $\mathbf{X}f$, $\mathbf{F}f$, $\mathbf{G}f$, or $f\mathbf{R}g$ with f, g path formulas.

The language we use for atomic propositions consists of medical actions ($Actions$), medical plans ($Plans$), and data structures ($Data$):

$$\begin{array}{ll} Actions: & \{\text{tumour-excision, mastectomy, AND, SNP}\} \\ Plans: & \{\text{TREATMENT, BCT, MRM, AXILLA-STAGING}\} \\ Data: & \{\text{CI-BCT, CI-SN, TF, SN, ITC}\} \end{array}$$

where $\text{CI-BCT}, \text{CI-SN} \in \{\top, \bot\}$ denote the contra indications for BCT and SN, respectively, $\text{SN} \in \{unknown, neg, pos\}$ denotes whether there is a metastasis found in the lymph nodes after performing the SN procedure, $\text{TF} \in \{unknown, \top, \bot\}$ denotes whether the resection margins are tumour free, and $\text{ITC} \in \{\top, \bot\}$ denotes whether there are isolated tumour cells. In the formal analysis, we will only concern ourselves with the surgical part of the treatment and omit radiotherapy.

3. Formalisation of Medical Management

In this section, we give a constraint-based representation of a fragment of the CBO guideline using the temporal logic representation discussed in the previous section. Furthermore, we interpret the recommendations in the IKO protocol and represent them in a more or less executable model. The goal is to verify whether the model of the protocol complies with the recommendations of the CBO guideline, or if there are differences, using a model checking approach [5].

3.1. Constraint-Based Representation of the Guideline

The final representation in temporal logic of the medical management in the CBO guideline is shown in Figure 1.

Some constraints given by the guideline are not easily expressible in temporal logic, as they involve other modalities different from time, such as the preference for BCT over MRM and the preference for the SNP over axilla-dissection for staging the axilla. Other assumptions regarding the patient data are implicit in the guideline, e.g., the status of the resection margins, i.e., whether they are tumour free (TF) or not ($\neg$TF), only becomes known after excision of the tumour and the existence of metastasis (SN=pos or SN=neg) only becomes known after the SNP. Here we have chosen not to consider these more implicit constraints.

Constraints related to control structure
(1) $\mathbf{AG}(\text{TREATMENT} \rightarrow \mathbf{AF}(\text{BCT} \vee \text{MRM}))$
(2) $\mathbf{AG}(\textsc{ci-bct} \rightarrow \neg\text{BCT})$
(3) $\mathbf{AG}(\text{BCT} \rightarrow \mathbf{AF}(\text{AXILLA-STAGING} \vee \text{MRM}) \wedge \mathbf{AF}\ \text{tumour-excision})$
(4) $\mathbf{AG}(\text{MRM} \rightarrow \mathbf{AF}\ \textsc{and} \wedge \mathbf{AF}\ \text{mastectomy})$
(5) $\mathbf{AG}(\text{AXILLA-STAGING} \rightarrow \mathbf{AF}\ (\textsc{and} \vee \textsc{snp}))$
(6) $\mathbf{AG}(\textsc{ci-sn} \rightarrow \neg\textsc{snp})$
(7) $\mathbf{AG}(\text{tumour-excision} \rightarrow ((\textsc{tf} = \bot \rightarrow \mathbf{AF}\ \text{MRM}) \wedge (\textsc{tf} = \top \rightarrow \mathbf{AG}\ \neg\text{MRM}))$
(8) $\mathbf{AG}(\textsc{snp} \rightarrow (\textsc{sn} = pos \wedge \neg\textsc{itc} \rightarrow \mathbf{AF}\ \textsc{and}))$
(9) $(\mathbf{G}\neg\text{MRM}) \rightarrow \mathbf{AG}(\textsc{snp} \rightarrow \mathbf{AG}(\textsc{itc} \rightarrow \mathbf{AG}\neg\textsc{and}))$
(10) $\mathbf{AG}(\text{TREATMENT} \rightarrow (\textsc{ci-sn} \rightarrow \mathbf{AF}\ \textsc{and}))$
Constraints related to data
(11) $(\textsc{ci-bct} \rightarrow \mathbf{AG}\ \textsc{ci-bct}) \wedge (\neg\textsc{ci-bct} \rightarrow \mathbf{AG}\ \neg\textsc{ci-bct})$
(12) $(\textsc{ci-sn} \rightarrow \mathbf{AG}\ \textsc{ci-sn}) \wedge (\neg\textsc{ci-sn} \rightarrow \mathbf{AG}\ \neg\textsc{ci-sn})$

Figure 1. Constraint-based representation of the CBO guideline. BCT = breast conserving treatment, MRM = modified radical mastectomy, CI-BCT = contra indications for BCT, SN = result of sentinel node procedure, CI-SN = contra indications for SNP, tumour-excision = segmental tumour excision, TF = tumour free resection margins, ITC = isolated tumour cells.

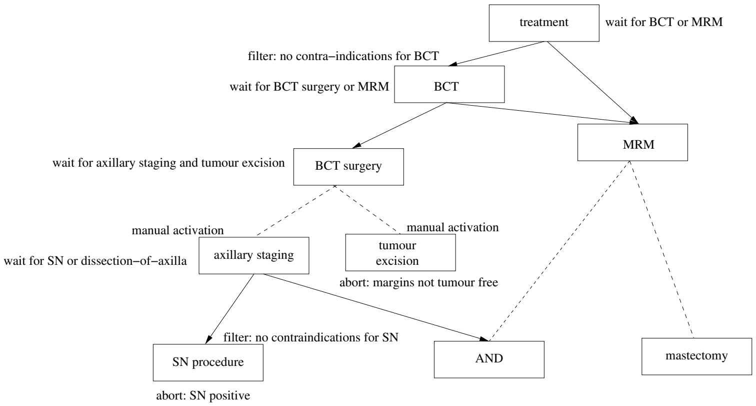

treatment
wait for BCT or MRM
filter: no contra–indications for BCT
wait for BCT surgery or MRM
BCT
MRM
wait for axillary staging and tumour excision
BCT surgery
manual activation
manual activation
wait for SN or dissection–of–axilla
axillary staging
tumour excision
abort: margins not tumour free
filter: no contraindications for SN
SN procedure
AND
mastectomy
abort: SN positive

Figure 2. Asbru interpretation of IKO protocol. Arrows represent sequential plans, dotted lines represent unordered sub-plans

3.2. Asbru Representation of the IKO Protocol

Much research has already been devoted to the development of representation languages for medical guidelines. Most of them consider guidelines as a composition of actions, controlled by conditions. However, many languages are not formal enough for the purpose of our research as they often incorporate free-text elements which do not have a clear semantics. Exceptions to this are PRO*forma* [7] and Asbru [15]. Here we use Asbru, because in previous research its semantics has been defined precisely [1] and can be translated automatically into SMV for model checking purposes [2].[3]

[3] `http://www.cis.ksu.edu/santos/smv-doc/` [accessed January 2008]

The overall structure of the Asbru model is given in Figure 2. It consists of nine plans ordered in a hierarchy. The top level plan 'treatment' will start by selecting the BCT plan, which may be rejected in case there are contra indications against doing breast conserving therapy. In that case, treatment will continue with a modified radical mastectomy (MRM). In case BCT is successfully completed, the treatment also completes and the MRM plan will not be started. The 'BCT surgery' plan consists of axillary staging and tumour excision, which are modelled as unordered plans, as the protocol does not explicitly state an order. To allow for a specific ordering of these two sub-plans we include a manual activation and assume that the activation will be performed by a doctor eventually. In case of BCT, the axillary staging starts with an investigation of the sentinel nodes provided that there are no contra indications. In case it is rejected or, because the sentinel nodes are positive, the plan is aborted, and an axillary dissection has to be performed. Furthermore, it is possible that the excision aborts because the margins are not tumour free. Since BCT surgery waits for this sub-plan, in that case BCT surgery has to be aborted and therefore it is mandatory to do a MRM. Finally, MRM is defined as both a dissection of the axilla and a mastectomy as defined by the protocol. No particular order between the two is given.

The formal semantics of the Asbru model in Figure 2 is based on the plan state model described in [1], of which an SMV model was constructed using the method and tool described in [2]. Most variables dealing with patient data are initialised as *unknown* and receive an indeterministic value in the second step to make sure there is only one root of the model. Furthermore, we assume that they do not change during the treatment. The only variables that are initialised at a later stage are the status of the sentinel node, which becomes known during the SNP and whether or not the tumour margins that have been resected are tumour free, which becomes known at the excision of the tumour. Furthermore, fairness constraints have been added to ensure that the manual activation of both the axilla surgery and the tumour excision eventually occurs. In other words, the patient will not wait indefinitely for the treatments to start.

Using the above formalisation, the IKO protocol can be verified using the constraints of the CBO guideline using standard model checking techniques. However, guidelines and protocols are usually under-constrained, thereby allowing many treatment paths not occurring in medical practice. We therefore look at the inclusion of medical management in practice in the following section.

4. Comparison Using Background Knowledge

In this section, we use the textbook of Roses [14] to create a precise model of medical management in practice. This model will be formalised into a decision tree, referred to as *background knowledge*, which will be used to select the part of the IKO protocol that is consistent with [14] and then verify for only this selected part whether it complies with the constrained based representation of the CBO guideline.

4.1. Medical Management in Practice

According to [14], the sentinel node procedure (SNP) is started before segmental excision (i.e., used in BCT) or mastectomy. The sentinel nodes (SNs) are then immediately sent to the pathology lab, where they are examined during surgery. If the SNs are found to be positive, axillary dissection can be completed during the primary breast surgery in one setting.

Furthermore, [14] differs from the CBO guideline and IKO protocol in the case of recurrent tumour positive resection margins in the BCT treatment. Whereas CBO and IKO recommend to switch the treatment to MRM, which includes axillary dissection, [14] only recommends a mastectomy with axillary dissection depending on sentinel node histopathology.

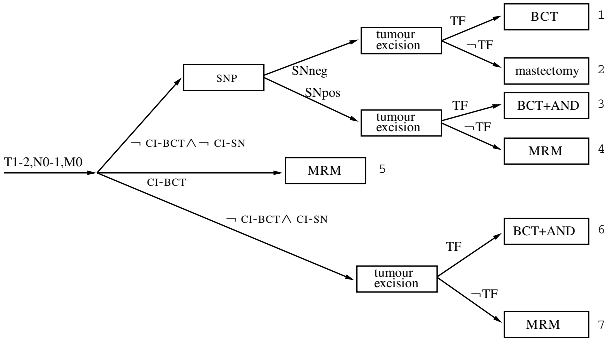


Figure 3. Background knowledge: possible treatment paths for surgery of operable invasive breast cancer. CI-BCT = contra indications BCT, CI-SN = contra indications SNP, TF = tumour free resection margins, AND = axillary node dissection.

Information from [14] can be represented in a decision tree as shown in Figure 3, which deals with the ordering of medical actions treating the primary tumour (BCT and MRM) and the axilla (SNP and/or AND).[4] Nodes represent medical actions or plans, arcs represent constraints. A path from the root node to a leaf node represents a treatment path, which defines the order of medical actions when the constraints on the path are satisfied. As mentioned above, the guideline recommends MRM in path (2) instead of mastectomy.

4.2. Comparing Medical Management with the IKO Protocol

Clearly, the medical management stated by the IKO protocol is less precise than the medical management performed in practice. Typically, one would expect the medical management in the protocol to be under constrained when compared to the medical management in practice. To verify this for the IKO protocol, we have transformed the 7 possible treatment paths in Figure 3 into a number of CTL properties (2 shown below) and verified whether these paths occur in the protocol.

$$\begin{aligned}
&(1)\ \mathbf{EX}(\neg\textsc{ci-bct} \wedge \neg\textsc{ci-sn} \wedge \mathbf{EF}(\textsc{snp} \wedge \textsc{snp} = neg \wedge \\
&\qquad\quad \mathbf{EF}(\text{tumour-excision} \wedge \textsc{tf} \wedge \mathbf{AG}(\neg\text{mastectomy} \wedge \neg\textsc{and})))) \\
&(2)\ \mathbf{EX}(\neg\textsc{ci-bct} \wedge \neg\textsc{ci-sn} \wedge \mathbf{EF}(\textsc{snp} \wedge \textsc{snp} = neg \wedge \\
&\qquad\quad \mathbf{EF}(\text{tumour-excision} \wedge \neg\textsc{tf} \wedge \mathbf{EF}(\text{mastectomy}) \wedge \mathbf{AG}(\neg\textsc{and}))))
\end{aligned}$$

[4] We abstract from isolated tumour cells.

With the SMV model checker we were able to verify that all paths, except (2), can occur in the IKO protocol. Path (2) does not hold in the IKO protocol because it recommends a MRM whereas [14] recommends a mastectomy, i.e., axillary dissection is included in the medical management according to the protocol, but not according to [14]. Whether the protocol or the textbook is incomplete or incorrect should be discussed with medical experts.

4.3. Selective Comparison of Guideline Constraints and Protocol

Clearly medical management is much less precisely defined in the CBO guideline and the IKO protocol than in the medical textbook of Roses [14]. Hence, any model that is only based on a written document of a guideline or protocol without the inclusion of background knowledge, will include many paths in which medical actions are unrealistically ordered. Many rightful properties of medical management may therefore not hold for the model constructed. Either one can choose to improve the model such that it adheres to medical practice (but not to the guideline document), or one can select only those paths in the model that also occur in medical practice for which then the property needs to be proven.

One approach to accomplish this is by including assertions to the model of the protocol or guideline. Assertions are statements that should hold in every execution path of the protocol, which, in SMV, are written down in the form of linear time logic (LTL) properties. This makes it possible to state properties about the relation of medical actions in time. In order to do this, the background knowledge formalised in terms of a decision tree, needs to be interpreted in terms of such LTL assertions. Here, we consider the following LTL assertions.

(1) $(\neg\textsc{ci-bct} \wedge \neg\textsc{ci-sn}) \leftrightarrow \mathbf{F}\,\textsc{snp}$
(2) $(\mathbf{F}\,\textsc{snp}) \rightarrow ((\neg\text{tumour-excision}\ \mathbf{U}\ \textsc{snp}) \wedge \mathbf{F}\ \text{tumour-excision})$
(3) $((\mathbf{F}\,\textsc{snp}) \wedge \textsc{snp} = neg \wedge (\mathbf{F}\,\textsc{tf})) \rightarrow \mathbf{G}(\neg\text{mastectomy} \wedge \neg\textsc{and})$
(4) $((\mathbf{F}\,\textsc{snp}) \wedge \textsc{snp} = neg \wedge (\mathbf{F}\,\neg\textsc{tf})) \rightarrow ((\mathbf{F}\ \text{mastectomy}) \wedge (\mathbf{G}\,\neg\textsc{and}))$
(5) $((\mathbf{F}\,\textsc{snp}) \wedge \textsc{snp} = pos \wedge (\mathbf{F}\,\textsc{tf})) \rightarrow ((\mathbf{F}\,\textsc{and}) \wedge (\mathbf{G}\,\neg\text{MRM}))$
(6) $((\mathbf{F}\,\textsc{snp}) \wedge \textsc{snp} = pos \wedge (\mathbf{F}\,\neg\textsc{tf})) \rightarrow \mathbf{F}\ \text{MRM}$
(7) $(\textsc{ci-bct} \rightarrow ((\mathbf{G}\,\neg\text{tumour-excision}) \wedge (\mathbf{F}\ \text{MRM})))$
(8) $(\neg\textsc{ci-bct} \wedge \textsc{ci-sn}) \rightarrow \mathbf{F}\ \text{tumour-excision}$
(9) $(\neg\textsc{ci-bct} \wedge \textsc{ci-sn} \wedge (\mathbf{F}\,\textsc{tf})) \rightarrow ((\mathbf{F}\,\textsc{and}) \wedge (\mathbf{G}\,\neg\text{MRM}))$
(10) $(\neg\textsc{ci-bct} \wedge \textsc{ci-sn} \wedge (\mathbf{F}\,\neg\textsc{tf})) \rightarrow \mathbf{F}\ \text{MRM}$

Assertions (1) and (2) deal with the use of sentinel node procedure and the order between this and the excision of the tumour. Assertions (3) to (6) are concerned with paths (1) to (4). Assertion (7) deals with path (5). Finally assertions (8) and (9) deal with path (6) and (7). However, we have seen in the previous section that (4), which corresponds to path (2), is not coherent with the model (i.e., from (4) it follows the antecedent of (4) is false), so in this form it is not usable. We could therefore either adapt the assertions so that it corresponds to the guideline or omit it. Here, we have omitted it.

Verifying the guideline constraints with SMV on the Asbru model of the IKO protocol using these assertions, shows that, indicating a difference between protocol and guideline with respect to medical management in practice. Although, in this case the difference between protocol and guideline is clear and could also have more easily been found through an informal analysis, this is largely because the protocol and guideline have a very similar structure and their recommendations are almost identical. However, the approach taken is independent of the underlying structure of the protocol and guideline. Therefore, this case study shows that formal techniques can be used to compare guideline and protocol independent of their underlying document structure.

5. Discussion

The aim of this work was to obtain insight into the differences and similarities between guidelines and protocols, based on the assumption that protocols should be looked upon as local modifications of guidelines. As a guideline is a starting point for drafting a protocol concerning the same topic, the development of a protocol based on a guideline can been seen as a transformation process. In this work, we have only been able to find end point protocols; as a consequence, the transformation process could only be described as consisting of a single step. In reality, it may be a more iterative process to design a protocol on the basis of an available guideline. This view on both guidelines and protocols raises a number of issues currently not addressed in literature.

First, guidelines are typically under-constrained thereby omitting many details about treatment order. Our work contrasts on this point with [16] for example, in which guidelines are more viewed as programs, but in which no execution paths are excluded that are illogical for medical management in practice. Clearly, additional medical background knowledge is needed to supplement the knowledge in the guideline document as was already acknowledged in previous work [9]. Whereas in previous work we incorporated background knowledge into the model, here we have used background knowledge to restrain the number of possible execution paths.

Second, researchers have focused on the verification of the quality of medical guidelines. However, verification of guidelines still takes a lot of effort. By using formal methods to find differences between a protocol and a guideline, one could reuse verification results of the guideline for the protocol and only focus on those parts that are different. Current work on verification of guidelines only considers guidelines to be solitary objects. No reference is made to verifying adaptations of guidelines.

Third, locating differences between guidelines and protocols is a novel topic, which has previously only been looked at from an informal angle [12]. The formal techniques used in our research extends previous work on model checking medical guidelines [2] and complements the techniques used in earlier work on quality checking medical guidelines [16,9]. The authors of [11] also investigate guideline-protocol relationships using model checking, but take a different approach as they view guidelines and protocols as programs rather than constraints on medical management.

One of the questions that emerged in the course of the research was whether the guideline or protocol ought to be adopted as the gold standard for comparison. Based on insights obtained by consulting literature on guideline development, we decided to take neither the guideline nor the protocol as the gold standard, but medical management in practice up to the point where it is consistent with the guideline and/or protocol. Using model-checking as principal tool, the guideline and protocol, now seen as defining logical constraints to medical management, were compared to a decision tree describing the medical management. Some of the outcomes of this research cast doubts on the content of both guideline and protocol, in the sense that at least some sort of explanation is needed in order to understand why there are differences between the decision tree, on the one hand, and the guideline and protocol, on the other hand. We believe that these results give a promising starting point for further investigating the relations between guidelines, protocols, and medical management in practice.

References

[1] M. Balser, C. Duelli, and W. Reif. Formal semantics of Asbru - an overview. In *Proceedings of the International Conference on Integrated Design and Process Technology*, Passadena, June 2002. Society for Design and Process Science.

[2] S. Bäumler, M. Balser, Dunets A., Reif W., and Schmitt J. Verification of medical guidelines by model checking – a case study. In *Proceedings of 13th International SPIN Workshop on Model Checking of Software*, LNCS. Springer-Verlag, 2006.

[3] M. Ben-Ari, Z. Manna, and A. Pnueli. The temporal logic of branching time. *Acta Inform*, 20, 1983.

[4] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Logic of Programs: Workshop*, number 131 in LNCS, Yorktown Heights, NY, May 1981. Springer.

[5] E.M. Clarke, O. Grumberg, and A.D. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, London, England, 2001.

[6] M. Field and K. Lohr, editors. *Clinical Practice Guidelines: Directions for a New Program*. National Academy Press, Institute of Medicine, Washington D.C., 1990.

[7] J. Fox, N. Johns, A. Rahmanzadeh, and R. Thomson. PROforma: a general technology for clinical decision support systems. *Computer Methods and Programs in Biomedicine*, 54:59–67, 1997.

[8] F.L. Green, D.L. Page, I.D. Fleming, A. Fritz, M.C. Balch, D.G. Haller, and M. Morrow. *AJCC Cancer Staging Manual*. Springer-Verlag, New York, 2002.

[9] A. J. Hommersom, P. C. Groot, P. J. F. Lucas, M. Balser, and J. Schmitt. Verification of medical guidelines using background knowledge in task networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(6):832–846, 2007.

[10] M. Marcos, M. Balser, A. Teije, and F. Harmelen. From informal knowledge to formal logic: A realistic case study in medical protocols. In *Proceedings of EKAW*, pages 49–64. Springer, 2002.

[11] M. Marcos, B. Martínez-Salvador, M. Balser, J. Schmitt, A. Dunets, S. Bäumler, and W. Reif. A formal approach to the refinement of medical practice guidelines into protocols. *Submitted to Computers in Biology and Medicine*, 2006.

[12] M. Marcos, B. Martínez-Salvador, A. Hommersom, P. Groot, P. Lucas, A. Jovell, and S. Blancafort. Case-study in transformations for protocol development from guidelines. Technical Report D5.1, Protocure II, 2006.

[13] A. Pnuelli. A temporal logic of concurrent programs. *Theoretical Computer Science*, 13:45–60, 1981.

[14] D.F. Roses. *Breast Cancer*. Elsevier, Philadelphia, PA, 2nd edition, 2005.

[15] Y. Shahar, S. Miksch, and P. Johnson. The Asgaard project: A task-specific framework for the application and critiquing of time-orientied clinical guidelines. *Artificial Intelligence in Medicine*, 14:29–51, 1998.

[16] A. ten Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen, P. Lucas, S. Miksch, W. Reif, K. Rosenbrand, and A. Seyfang. Improving medical protocols by formal methods. *Artificial Intelligence in Medicine*, 36(3):193–209, 2006.

[17] J.J.E. van Everdingen, J.S. Burgers, and W.J.J. Assendelft. *Evidence-based richtlijnontwikkeling*. Bohn Stafleu van Loghum, 2004.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-223

TSNet – A Distributed Architecture for Time Series Analysis

Jim HUNTER[1]
Department of Computing Science
University of Aberdeen, UK

Abstract. This paper describes an infrastructure (TSNet) which can be used by geographically separated research groups to develop algorithms for the abstraction of complex time series data. The framework was specifically designed for the kinds of abstractions required for the application of clinical guidelines within intensive care.

Keywords. time series, intensive care, data abstraction, clinical guidelines, distributed architectures

Introduction

Abstracting complex time-series data is a necessary precursor to the application of clinical guidelines and protocols in areas such as the Intensive Care Unit where large volumes of data are generated. If we are to obtain a consensus as to the best ways to achieve these abstractions, different research groups need to be able to experiment with data acquired from a variety of sources, and to apply algorithms developed elsewhere. However, sharing of data and algorithms is known to be difficult. This paper sets out an architecture which has been developed to allow collaborative working of this kind.

1. Background

Intensive and high-dependency care is becoming increasingly complex. Up to ten physiological parameters (such as heart rate, oxygen and carbon dioxide levels in the blood, body temperatures and blood pressures) can be continuously monitored as frequently as once per second. In addition to this continuous physiological data, there will be a number of items which are entered sporadically – laboratory results, blood gases, medication, equipment settings, patient observations, etc. Imaging apart, an Intensive Care Unit (ICU) generating almost a million measurements per patient per day, is arguably the clinical environment that generates the largest volumes of data.

Medical errors in the ICU are not uncommon, and although the majority are unimportant and thus might not be classed as 'mistakes', some are significant causing dete-

[1] Corresponding author: Jim Hunter, Department of Computing Science, University of Aberdeen, King's College, Aberdeen AB24 3UE, UK; Email: j.hunter@abdn.ac.uk

rioration of the patient and acute anxiety for both relatives and staff. Errors often result from missed symptoms and signs, or a lack of appreciation of their importance. Intensive care is an extraordinarily complicated environment and things get forgotten.

Many units have addressed this problem by summarising the data using complex graphical displays. While this is of significant help to senior clinicians, it has not been shown to improve the performance of the junior staff [1].

One way in which junior staff might benefit from the availability of these data, would be to automatically apply clinical guidelines to the data for an individual patient in order to draw their attention to the recommendations for that patient when appropriate. However, guidelines will often be expressed using abstractions which are not directly available in the raw data and a variety of signal processing algorithms (both numerical and knowledge-based) must be applied. For example the abstraction: *multiple bradycardias occurring over several minutes* must be derived from a sequence of real valued measurements of heart rate.

This process of abstraction may involve time series from only one numerical variable or from several; it may also involve data which are entered sporadically (e.g. laboratory measurements). It will almost certainly involve the removal of low level artifact arising from patient movement, ambient noise and clinical intervention [2]; higher level abstractions may involve segmentation [3], trend detection, Markov modelling [4] and other sophisticated pattern matching techniques.

Another requirement for the automatic application of guidelines is an execution engine (interpreter) which will apply the formalised guideline to the data abstractions and return recommendations to the user.

At present, most of the above abstraction and guideline execution techniques are being developed by individual research groups, who apply them to data generated by their own clinical collaborators. There is little sharing either of data or of algorithms.

In part, this is due to the lack of a suitable infrastructure to enable such sharing to take place. This paper presents a suitable infrastructure which allows sharing across the internet. Our vision is of a researcher in group A, being able to access data acquired by group B, and comparing a signal processing algorithm she has developed (say in MatLab) with algorithms developed by groups C (written in Java) and D (written in Delphi) and running a guideline written by group E on an interpreter provided by group F.

Collaboration is necessary because:

- The abstraction of complex ICU time series data is difficult; we do not know in advance which approach(es) will bear fruit.
- People tend not to appreciate the advantages or disadvantages of algorithms developed by others until they have tried them themselves.
- It is dangerous to claim generality for a specific approach until it has been tested on data from different sources.

The basic architecture we have implemented (called TSNet) is, unsurprisingly, based on the standard client/server model. In section 2, we will set out a motivating example; section 3 defines some terms used in TSNet and section 4 describes its architecture and how it would be used on the motivating example. Finally, section 5 provides a comparison with similar systems and conclusions.

2. Example

Consider a neonate who is being artificially ventilated until her lungs are sufficiently mature for her to breathe on her own. Often the immaturity manifests itself in the condition known as respiratory distress syndrome (RDS). The aim of our (almost trivial) protocol is to issue recommendations for maintaining suitable values of the oxygen (O_2) in the blood as measured by a single transcutaneous probe. A number of ventilator settings, under the control of the nurse, can be used to provide this regulation but we will assume that he uses the percentage of oxygen in the inspired air (FiO_2) to control the blood O_2. For further details of this protocol and its translation into Asbru, see [5].

We implemented the advisory system using the following processing modules: a raw data source (from the Neonate project [6]); a median filter to smooth the O_2 data; a means of removing artifacts (the ArtiDetector [2]); an Asbru interpreter; a display to present the data to the user. The network is shown in Figure 1.

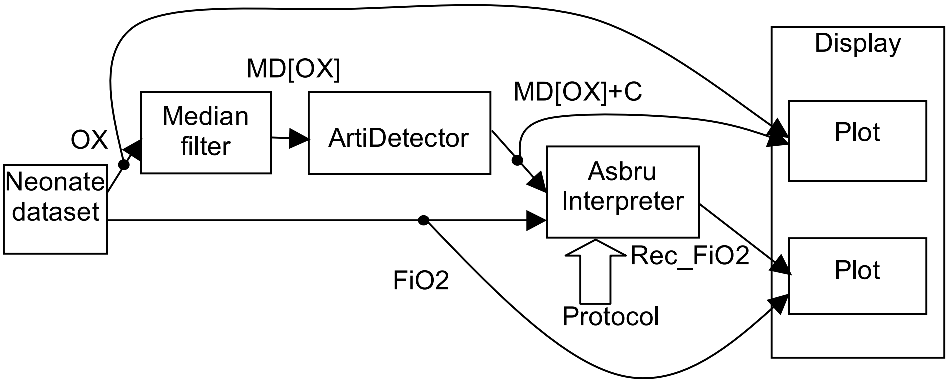


Figure 1: Processing and display modules

An example of the application of this network to a real data stream is shown in Figure 2; the top plot shows blood O_2 (OX): the faint trace shows the raw data and the bold trace the output of the ArtiDetector(MD[OX]+C). The second graph presents FiO_2: the solid trace shows the actual values and the dotted trace the recommendations of the protocol.

3. TSNet Definitions

Before showing how we distribute the elements of this example in TSNet, we need to define some essential concepts.

3.1. Channels

A *channel* consists of a named data stream. The data may be the raw data, or may be the result of some form of processing. Channels have two main sub-classes – equi-sampled and interval. The data values in *equi-sampled* channels are (as the name implies) acquired at a regular, constant frequency. The data 'values' can be a variety of different types: numerical (floating point); Boolean; enumerated (in the Pascal sense – i.e. sequential integers starting at 0); a vector of floating point values – typically a frequency spectrum.

An *interval channel* consists of a set of temporal intervals, each defined by: start and end date/times; an attribute name; a value (of any type). Intervals can be of zero length duration, known as *events*. Interval attributes can be organised into a tree structure, known as a *descriptor tree*. Irregularly sampled numerical channels can also be handled by an interval channel of events.

In our example, OX, MD[OX] and MD[OX]+C are equi-sampled numerical channels; FiO2 and Rec_FiO2 are interval channels.

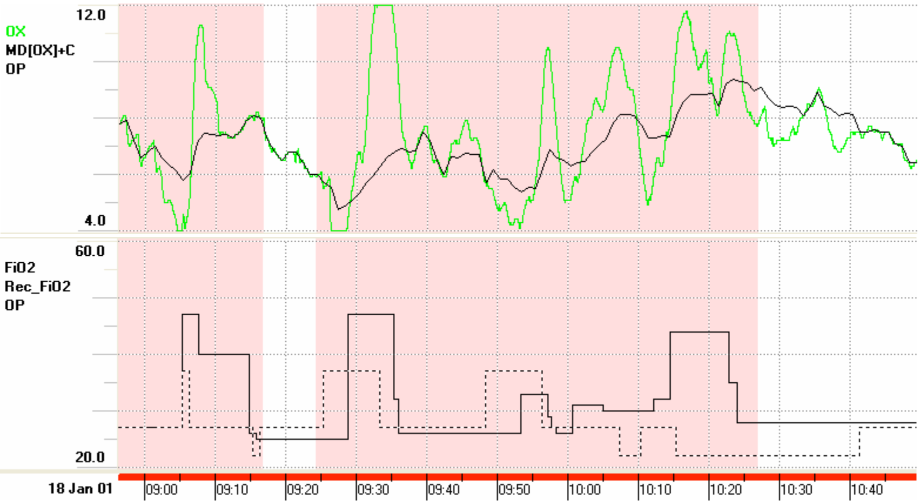


Figure 2: Data and recommendations

3.2. Filters

Within TSNet, any module which has zero or more input channels and zero or more outputs channel as is known as a *filter*. A very simple example would be the family of moving window filters (mean, standard deviation, median, slope, etc.) which take in one equi-sampled numerical channel and output a channel of the same class.

A special sub-class of filter is the *data source*, which has no inputs but whose output channels provide the raw data from a particular source; of course there are inputs in the forms of files or databases, but the function of the raw data source is to hide the details of the format of these data from the rest of the system. Another sub-class is the *data sink*, which has one or more input channels but no output channels. The prime example is the *plot*, where input channels are converted to a visual representation. The most common type of plot is that of the values of a variable against time, but different types of plot are appropriate for representing interval channels, descriptor trees, spectra, etc.

The introduction of sinks and sources means that every channel is the output of some filter and acts as the input to one or more filters.

In our example, the median filter and the ArtiDetector are filters and the Neonate dataset is a data source.

Filters can carry out more complex operations such as segmentation, where the filter takes in an equi-sampled numerical channel and outputs an interval channel - the intervals representing the segments. Filters can even be complex rule-based pattern recognisers, one example being a filter that recognises the presence of a transcutaneous probe change [7]. Another example is the Asbru Interpreter which takes in a number of processed channels (with artifacts removed) and outputs an interval channel containing the recommendations which result from the application of a particular guideline written in Asbru [8].

One barrier to collaboration between groups is that they may well use different programming languages and are reluctant to devote time to translating their algorithms into languages that other groups can easily use. By defining standards for channels and for the interfaces to filters, TSNet enables filters to be written in a variety of languages including Java, Delphi, MatLab, R and CLIPS.

An output channel of one filter can be an input to another thus enabling the construction of complex channel/filter *networks*. Filters specify the classes of channel that they input and output and any network must respect these type constraints. A plot specifies its input channels and a channel specifies the filter it is derived from (which in turn will specify other channels and filters).

3.3. Contexts and Catalogues

TSNet is designed to be flexible and incremental. To this end, *contexts* are defined and organised hierarchically. Typically a context corresponds to a specific data source and/or project. Contexts provide an environment for the definition of channels, filters (especially raw data sources) plots and displays. All contexts inherit from the ROOT context. As is usual with inheritance hierarchies, the advantage is that filters, channels etc. are inherited down the hierarchy. Filters which are used extensively (e.g. involving moving windows or segmentation) can be defined in the ROOT context, whereas more specialised filters can be defined at an appropriate level. An example of a deep hierarchy is found in the Neonate project [9, 10].

A *catalogue* is a list of time periods (called data periods) which the researcher wishes to study. Contexts can have as many catalogues defined for them as is desired. The catalogue is displayed to the user for her to select a data period for processing and display.

3.4. TSNet Classes and Components

The TSNet architecture makes explicit the classes which are available and which can be referred to – these are implementation independent. The base class is TSNetComponent and channels, filters, plots, contexts and catalogues are all sub-classes of this. All TSNetComponents can have associated *parameters* which can be of type string (with several specialised variants), integer, float and Boolean. In our example, the moving window median filter needs to know the width of the window, and the amount by which it is advanced.

4. TSNet Architecture

4.1. Use Cases

We will illustrate the way in which the TSNet distributed architecture operates with a number of use cases based on the network shown in Figure 1.

Firstly, a user in group ***α*** develops a filter class (in this case the Asbru Interpreter) keeping it private. When she is satisfied with it, she makes it publicly available by providing it as a web service on her group's server; TSNet provides a standard wrapper for filters implemented as web-services. She also loads a description of the filter on to the central configuration server; this description includes the types of the input and output channels, and the parameters used (including the URL of the protocol to be interpreted).

Then a user in a different group (***β***), makes the Neonate dataset publicly available, again using a standard TSNet wrapper, and loads a description of the dataset (including the channels that it generates, their types and parameters, etc) on to the central configuration server.

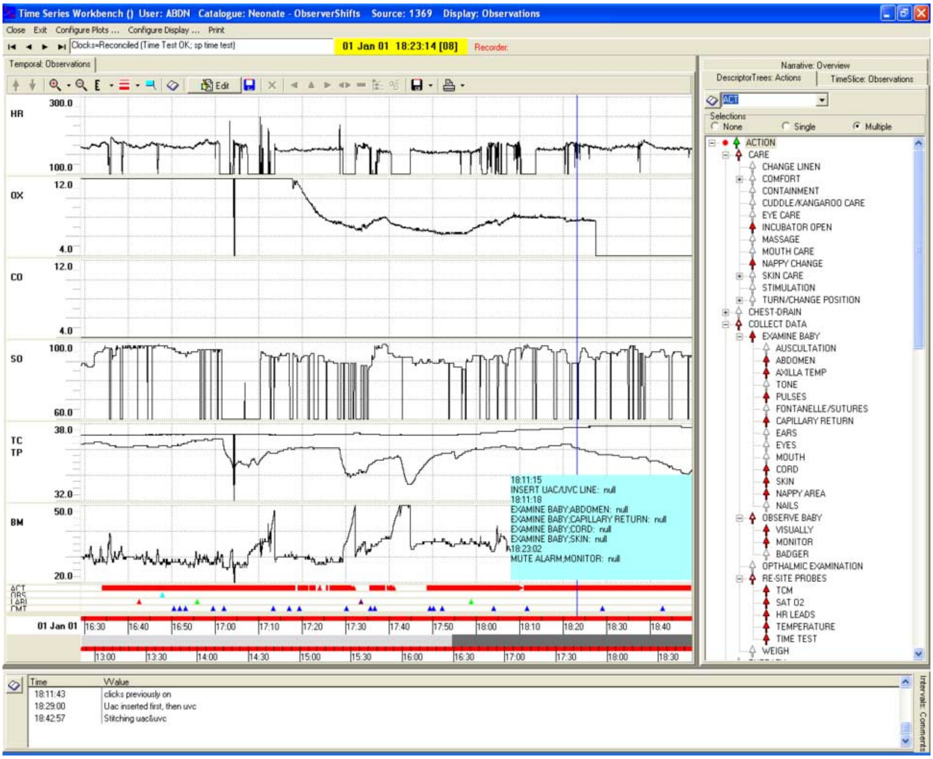
Sometime later a user in group ***δ*** wishes to develop a network to provide advice on FiO_2 regulation; she recognises that the signals used by the Asbru Interpreter need to have artifacts removed and develops the ArtiDetector on her client; we will assume that a median filter already exists within her client. She wants to test the network on the Neonate dataset. She firstly constructs the network on her client, using TSNet to import information about the dataset and the interpreter from the central configuration server, and saves the network locally. She then selects a data period from a catalogue of the Neonate dataset and executes the network. TSNet retrieves the selected data channels (OX and FiO2) from group ***α*** 's server, passes OX though the local median filter and ArtiDetector, sends the resulting channel (MD[OX]+C) and the FiO2 channel to group ***β*** 's implementation of the Asbru interpreter, retrieves the resulting channel (Rec_FiO2) and plots it, along with various other channels on her client (see Figure 2).

She finally decides that the FiO_2 regulator network she has developed may be of interest to others, and makes a description of it available on the central configuration server.

4.2. TSNet Clients

From the foregoing description, it can be seen that TSNet clients enable the user (i) to configure channels, filter classes, and plots and (ii) to display the data from a selected data period. Clients may implement filters internally or they may invoke external filters as web-services. At present only one serious client, the Time Series Workbench (TSW) has been implemented (in Delphi); however a light-weight prototype has been written in Java to be downloaded and run under a standard web-browser.

Channel configuration includes the introduction of new raw data channels and channels derived from other filters, the deletion, copying and renaming of channels, the specification of the parameters of channels and of the filters from which they are derived.



18:11:15
INSERT UAC/UVC LINE: null
18:11:18
EXAMINE BABY;ABDOMEN: null
EXAMINE BABY;CAPILLARY RETURN: null
EXAMINE BABY;CORD: null
EXAMINE BABY;SKIN: null
18:23:02
MUTE ALARM;MONITOR: null

Time	Value
18:11:43	clicks previously on
18:29:00	Uac inserted first, then uvc
18:42:57	Stitching uac&uvc

Figure 3: TSW Display

The simplest *plot* in TSW is a graph of values against time (referred to as a *temporal* plot) but there are many other possibilities: the descriptor tree associated with a channel; interval plots can list the attributes and values of the intervals forming an interval channel; time slice plots show the values present in all channels at a particular point in time. Figure 3 shows a temporal plot (top left), a descriptor tree (top right) and an interval plot (bottom).

In configuring a *plot*, the user specifies which channels are to be presented and how. As with channels, plots can be created, deleted, copied and renamed.

Within the TSW, a *display* consists of a number of plots, laid out according to the wishes of the user; it consists of a number (currently three) of separate areas; see Figure 3. Each area can contain as many plots on selectable tabs as are required. Configuring a display means allocating plots to specific areas on the display.

The client manages the execution of a network in order to display the data specified by an individual data period. In order to improve efficiency, only those channels which are required for plotting are computed; this subset of all possible channels is derived by working backwards from the plots through the channel/filter network.

A vital parameter for any filter is *Execution at*; this defines whether the filter is implemented internally within the client or externally as a web service. It makes sense to implement some filters (especially simple ones which belong to the ROOT context) within the client for the sake of efficiency. In our example, the median filter would is implemented in the client for this reason.

4.3. Configuration Web Service

The main aim of TSNet is to allow collaboration between different research groups. This means that channels, filter classes, plots and displays can be named and described by one group or individual in such a way that they can be used by others. This requires that mechanisms be found for managing name spaces and for exchanging definitions.

Groups and individuals are organized in a hierarchy of *originators*. At the root of this hierarchy is a super-user called CORE. This user corresponds to the TSNet administrator, and is responsible for those entities which are available to all users. Such entities will belong to the ROOT context. Under CORE will be a number of groups and within each group a number of individual researchers.

The built-in TSNet classes were described in section 3; all of these classes belong to the ROOT context, are owned by the CORE originator and can not be altered. However subclasses of the filter class can be defined by other originators. Each filter subclass is described by: the name of the class; an ordered list of the input and output channel classes; a list of the parameters which are applicable to this filter class (in addition to those defined for the generic filter) and default values for those parameters. The originator also needs to provide a textual description of the semantics of the filter class; this is the only place where these semantics are set out - it is up to other users to decide whether the filter provides a functionality which is of use to them.

Instances of channels, filters, plots, displays and networks as configured by the user are held in the same way as the TSNet classes.

For efficiency, an individual user may have her own local configuration database managed directly by her client. Initially this will contain descriptions of those classes and instances which are built-in to her client, those belonging to the ROOT context and those which she has developed herself. She can also downloaded configurations defined by others from the central configuration server. Note that for filters she is not importing the filter code, but only sufficient information for her client to refer to the filter and to configure it into channel/filter networks. Likewise for datasets, in importing the configuration, she is not importing the data.

4.4. Filter Web Services

Currently TSNet is based on web-service architecture using Apache Tomcat. The primary interface on the server side is a filter manager which locates the filter (or dataset) concerned and arranges for the channel data to be acquired, presented to the filter and recovered from it, and transmitted back to the requester. Special care has been taken to optimize the structure of the SOAP messages to ensure the rapid transfer of large volumes of data.

Making filters available as web services has the advantage that the originator of the filter can make it available to the community without giving up ownership. We envisage that other levels of access could be made available, such as allowing the copying of the compiled code, or eventually of its source.

It is possible for a site to expose the description of a complex filter (i.e. one composed of a sub-network of other filters) while retaining the knowledge about its internal workings. Thus an external user of such a complex filter sends the input channels to the site which ‘owns’ the filter. That site then takes responsibility for managing the passage of the data through the elements of the complex filter, even when this may involve exporting the data to a third site which owns filters which contribute to the operation of the complex filter.

Confidentiality is important when it comes to exchanging medical data. TSNet assumes that all input data sets have been fully anonymised before being made available.

5. Comparisons and Conclusions

We are aware of relatively few attempts to create the sort of infrastructure we are proposing. TSNet has much in common with the MEDIATOR architecture [11] and specialisations thereof designed to handle time oriented data [12, 13]. What differentiates TSNet from these systems is its emphasis on the processing of large volumes of rapidly sampled data. The consequences of this emphasis may be implementational rather than conceptual, but they are none the less significant.

PhysioNet [14] provides both a data bank of physiological data (PhysioBank) and software to process and visualise the data (PhysioToolkit). PhysioNet differs significantly from TSNet in that (i) PhysioBank predominantly consists of ECG signals (although there are EEC and gait recordings) sampled at higher frequencies than those provided by monitors in the ICU; (ii) PhysioToolkit is a library of modules which the user downloads and integrates into her own application. However, before using this system, a UNIX emulator and additional software must be downloaded, installed, and configured on the client computer. Analysis of time-series data not in the PhysioNet data bank requires conversion of the data into a specific format, and storage and management of the data files always need to be done on the client's computer.

TSNet is most similar to Tempo [15]. The overall architecture of Tempo is built upon a specific data model and organized as pipelines of modules, assembled according to a data processing meta-model. Each pipeline module conforms to well-defined communication rules and wraps one or more data processing algorithms which can be delivered both as web-services and as software library.

The main focus of PAS (Physiology Analysis System) [16] is on physiological data collected during the transportation of patients from the site of injury to hospital, but the type of data is similar to that which would be collected in intensive care. PAS is similar to TSNet and Tempo in that data analysis is accomplished through a chain of 'functions' chosen from a pre-existing library; functions can be written in a variety of programming languages. The main difference is that PAS is a centralised system with data and algorithms being held in one location. Users access the system via a web browser.

Kepler [17] is a general purpose system for organizing scientific workflows; it includes the facility to conduct experiments through iteration and selection.

TSNet has been fully implemented and available for general use[2]. Its future development is expected to include a more detailed comparison with Tempo and their possible merging. We will also look at how Kepler might provide a more powerful network construction and workflow management facility. Issues of standardized vocabularies need to be addressed. The success or otherwise of TSNet will be established by the extent to which it is adopted by the user community for whom it is intended.

[2] http://www.csd.abdn.ac.uk/research/tsnet

Acknowledgements

The author is grateful to all of those who have contributed to the development of TSNet over the years, but in particular to Christian Fuchsberger, Steven McHardy, Lazaros Mavridis, Paul McCue, Ying Gao, Andrew Gardner and Neil Young. The NEONATE project was supported by the UK ESRC/EPSRC PACCIT Programme.

References

[1] S Cunningham, S Deere, A Symon, RA Elton and N McIntosh, "A Randomized, Controlled Trial of Computerized Physiologic Trend Monitoring in an Intensive Care Unit", *Crit.Care Med*, Vol 26, pp 2053-2060, 1998.

[2] CG Cao, IS Kohane, N McIntosh and K Wang, "Artifact Detection in the PO_2 and PCO_2 Time Series Monitoring Data from Preterm Infants"*, Journal of Clinical Monitoring and Computing*, **15**, pp 369-378, 1999.

[3] E Keogh, S Chu S, D Hart D, M Pazzani, "An Online Algorithm for Segmenting Time Series", *Proceedings of IEEE International Conference on Data Mining*, pp 289-296, 2001.

[4] CKI Williams, J Quinn, N McIntosh, "Factorial Switching Kalman Filters for Condition Monitoring in Neonatal Intensive Care", to appear in *Advances in Neural Information Processing Systems* 18, 2006.

[5] C Fuchsberger, JRW Hunter, P McCue, "Testing Asbru Guidelines and Protocols for Neonatal Intensive Care", AIME-05: Proceedings of the Tenth European Conference on Artificial Intelligence in Medicine, Springer Verlag, pp 101-110, 2005.

[6] JRW Hunter, L Ferguson, Y Freer, G Ewing, R Logie, P McCue and N McIntosh, "The NEONATE Database", *Workshop on Intelligent Data Analysis in Medicine and Pharmacology and Knowledge-Based Information Management in Anaesthesia and Intensive Care*, *AIME-03*, pp 21-24, 2003.

[7] JRW Hunter and N McIntosh, "Knowledge-Based Event Detection in Complex Time Series Data", AIMDM'99: Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, Horn W et al. (Eds.), Springer Verlag, pp 271-280, 1999.

[8] S Shahar, S Miksch and P Johnson, 'The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines', *Artificial Intelligence in Medicine*, **14**, pp29–51, 1998.

[9] JRW Hunter, Y Freer, G Ewing, R Logie, P McCue and N McIntosh, "NEONATE: Decision Support in the Neonatal Intensive Care Unit – A Preliminary Report", *AIME-03: Proceedings of the Ninth European Conference on Artificial Intelligence in Medicine*, Springer Verlag, pp 41-45, 2003.

[10] A Law, Y Freer, JRW Hunter, RH Logie, N McIntosh, J Quinn, "A Comparison of Graphical and Textual Presentations of Time Series Data to Support Medical Decision Making in the Neonatal Intensive Care Unit", *Journal of Clinical Monitoring and Computing*, 19, pp 183-19, 2005.

[11] G Wiederhold and M Genesereth, "The Conceptual Basis of Mediation Services", *IEEE Expert*, **12**(5), pp 38-47, 1997.

[12] J Nguyen et al., "Integration of Temporal Reasoning and Temporal-Data Maintenance into a Reusable Database Mediator to Answer Abstract Time-Oriented Queries: The Tzolkin System", *J Int. Inf. Sys.*, 13, pp 121-145, 1999.

[13] D Boaz and Y Shahar, "A Framework for Distributed Mediation of Temporal-Abstraction Queries to Clinical Databases", *Artificial Intelligence in Medicine*, **34**(1), 2005.

[14] AL Goldberger, LAN Amaral, L Glass, JM Hausdorff, PCh. Ivanov, RG Mark, JE Mietus, GB Moody, C-K Peng and HE Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation* 101(23):e215-e220, 2000.

[15] P Ciccarese and C Larizza, "A Framework for Temporal Data Processing and Abstractions", *Proceedings AMIA 2006*, pp146-150, 2006.

[16] TM McKenna, G Bawa, K Kumar and J Reifman, "The physiology analysis system: An integrated approach for warehousing, management and analysis of time series physiology data", *Computer Programs and Methods in Biomedicine*, **86**, pp 62-72, 2007.

[17] B Ludscher, I Altintas, C Berkley, D Higgins, E Jaeger, M Jones, E Lee, J Tao and Y Zhao, "Scientific Workflow Management and the Kepler System", *Concurrency and Computation: Practice & Experience*, p 36, 2005.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-233

Clinical Guidelines and Care Pathways: A Case Study Applying PRO*forma* Decision Support Technology to the Breast Cancer Care Pathway

Vivek PATKAR [a], [1] and John FOX [b]
[a] *Department of Academic Oncology, UCL, London*
[b] *Department of Engineering Science, Oxford*

Abstract. PRO*forma* is a formal language for modeling clinical processes that was developed by the Advanced Computation Laboratory of Cancer Research UK in 1996, together with associated tools, for creating decision support, care planning, clinical workflow management and other applications. The technology has been used to develop and evaluate a number of decision support applications in range of clinical settings. Clinical trials have been carried out and published for seven of these applications, all suggesting major positive benefits on a variety of outcome measures. The most recent and ongoing project called CREDO is an ambitious attempt to address the challenges in deploying sophisticated decision support systems in the intricate and convoluted management of chronic diseases, taking breast cancer as an example. In this chapter we describe the implementation of evidence based clinical guidelines within a complex care-pathway for patients with breast symptoms and analyse in detail the results of an evaluation study. Some important lessons learned during the process are shared and future directions are discussed.

Keywords. computerised decision support, clinical practice guideline, work flow.

Introduction

The growing influence of evidence based medicine has promoted worldwide interest in developing Clinical Practice Guidelines (CPGs) on the basis that they can be expected to help improve quality of care by disseminating research results and evidence based practice more effectively. The direct link between healthcare providers' adherence to quality guidelines and improvement in patient outcomes has been supported by many studies in the literature [1, 2].

There are, however, significant issues about the practical use of CPGs. There is a growing concern that the enormous effort that goes into creating the guidelines may not be matched by the level of adherence to them in practice and computer interpretable guideline is seen as a potential remedy to improve compliance. Though the principles behind computer executable CPGs are clearly promising the implementation of

[1] Corresponding Author: Clinical Research Fellow, Academic Oncology, Royal Free Hospital, NW3 2PF London, UK; E-mail: v.patkar@medsch.ucl.ac.uk

computer executable CPGs in complex clinical environments is a challenging task and the evidence in the literature about their effectiveness is still limited.

Our group at the Advanced Computation Laboratory of Cancer Research UK, has developed an executable guideline technology for supporting complex health care processes, patient management workflows and clinical decision-making. The technology includes the PRO*forma* [3] guideline modelling language, in which clinical processes are represented in terms of "tasks" like decisions, plans and clinical actions organised into processes, pathways, workflows etc. PRO*forma* has many features of the family of "task network" formalisms reviewed by Peleg et al 2004 [4]. A variety of decision support applications based on PRO*forma* technology have been developed and tested in clinical trials, and all these trials demonstrated significant positive impact: drug prescribing (CAPSULE) [5]; cancer referrals (ERA); HIV treatment (RETROGRAM) [6]; mammography interpretation (CADMIUM) [7]; leukaemia chemotherapy dosing (LISA) [8] and genetic risk assessment (RAGs) [9].

The most recent project called CREDO is intended to build on our existing experience and results by developing a system to support an entire breast cancer care pathway from the initial diagnosis through to treatment and follow-up (see video at [2]).

In this chapter we describe a decision support application developed within the CREDO project. This focuses on a key part of the breast care pathway commonly referred to as Triple Assessment. Our Triple Assessment Decision Support (TADS) system was developed with two primary aims. First, to investigate whether computer enacted guidelines can significantly enhance the compliance of breast clinicians with best practice, and second, to collect feedback from the clinicians on system usability.

1. Triple Assessment Decision Support System (TADS)

The knowledge required for the TADS system was modeled using the CommonKADS methodology described by Schreiber et al [10]. In the first phase we focused on analysis of the organisation in which TADS is to be used, describing the care processes, resources, and knowledge assets of the organisation and the expected impact of the CDS system on the organisation. Interviews were conducted with breast surgeons, pathologist and radiologist to gather inputs about clinical processes, workflows and functional requirements. In the second 'conceptual analysis' phase we abstracted the knowledge that the system is required to represent from conventional knowledge sources like national and international CPGs and medical literature, together with the reasoning that is to be performed using the knowledge, and the interactions with users and other external agents. In the third phase, the TADS system itself was constructed using the PRO*forma* decision and workflow modelling language [11] and the Tallis process modelling system[3].

[2] http://www.cossac.org/credoVideo.html

[3] http://www.cossac.org/tallis/index.html

1.1. Organisational Context and Workflow

In the UK, women with breast related symptoms that raise suspicion of breast cancer are referred by their GP to designated breast clinics called Triple Assessment (TA) clinics in local hospitals. The first element of the assessment consists of a clinical examination and the gathering of patient history and demographic data. If this initial examination reveals an abnormality then the patient will be referred for imaging and needle biopsy.

Four key inter-dependent decision points were identified in the TA workflow:

Risk assessment decision: Genetic risk assessment calculated as low, medium or high (taken as part of the clinical history plan.

Imaging decision: Radiological investigations to perform, for instance, mammogram, ultrasound, both or none.

Biopsy decision: Method to perform biopsy for instance, FNA, core biopsy and other forms of biopsies.

Management decision: Whether to refer the patient to a multi-disciplinary team (MDT) and/or to geneticist, to discharge or to follow up (high risk surveillance).

A simplified *PROforma* view of the triple assessment workflow within TADS is shown in Figure 1.

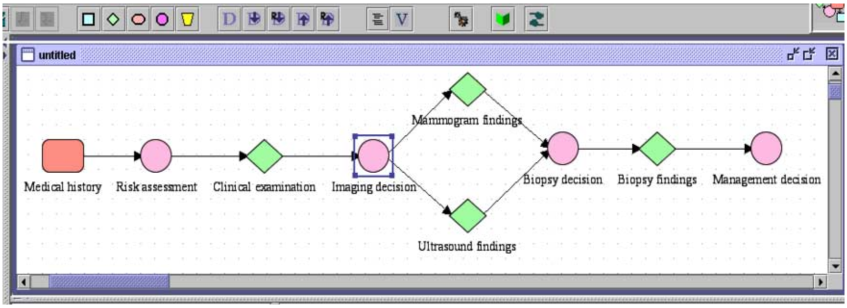
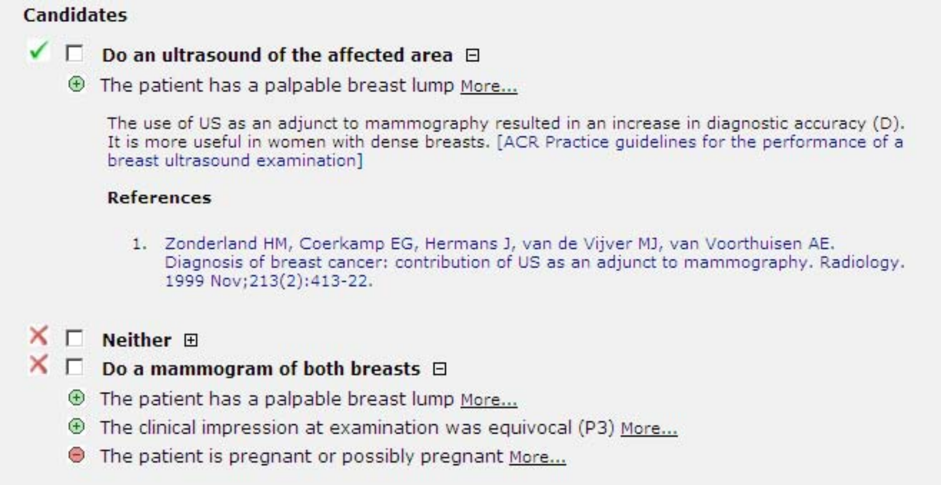


Figure 1. A simplified view of triple assessment as a task network (as displayed in the Tallis authoring tool). This shows the main plan which embodies a simple linear workflow with conditional branches depending on the imaging modalities used (mammography and/or ultrasound). The four decision nodes are represented by circles and are embedded at various points in the workflow.

1.2. The TADS Decision Model

A key feature of the TADS system is the argumentation approach used in the PRO*forma* decision model. This provides a clinically intuitive solution to the task of identifying and presenting relevant clinical evidence relating a specific patient state to a guideline recommendation (see Figure 2). A set of arguments '*for'* and '*against'* each patient management option is generated by the TADS decision engine, and the arguments are aggregated into a summary yielding a preference ordering over the options. Depending upon the overall assessment of benefit or harm the preferred management option is either displayed as '*recommended'* (indicated with a green tick) or '*non-recommended*' (red cross). The user can also see the medical reasons for and against each option, along with hyperlinks to the source CPG and underlying research evidence. Note, however, that the clinical user always has the freedom to override system recommendations.

Another advantage of the argumentation approach is in helping the user to take properly informed decisions by exposing any conflicts in the underlying evidence. This is not an uncommon scenario in medicine due to its inherent uncertainties and constant publication of new research findings or analyses.



Candidates

✓ □ **Do an ultrasound of the affected area** ⊟

⊕ The patient has a palpable breast lump More...

The use of US as an adjunct to mammography resulted in an increase in diagnostic accuracy (D). It is more useful in women with dense breasts. [ACR Practice guidelines for the performance of a breast ultrasound examination]

References

1. Zonderland HM, Coerkamp EG, Hermans J, van de Vijver MJ, van Voorthuisen AE. Diagnosis of breast cancer: contribution of US as an adjunct to mammography. Radiology. 1999 Nov;213(2):413-22.

✗ □ **Neither** ⊞

✗ □ **Do a mammogram of both breasts** ⊟

⊕ The patient has a palpable breast lump More...

⊕ The clinical impression at examination was equivocal (P3) More...

⊖ The patient is pregnant or possibly pregnant More...

Figure 2. TADS screen showing decision options for the imaging for one case, to be taken after medical history and examination. The system recommends an ultrasound scan but recommends against mammography and against doing nothing. For the decision option 'Do an ultrasound of the affected area', argument has been expanded to show the justifying evidence (an option available to the clinician for all decisions, options and arguments). Links are provided to the relevant supporting literature, which can be accessed by the user if required (e.g. from PubMed)

1.3. PROforma Technology and Guideline Authoring Toolkit

PRO*forma* is a knowledge representation language for describing decisions within processes that unfold over time, and may require the cooperation of various actors, such as clinicians or other medical personnel. Although declarative the task based structure is amenable to a form of visual programming which facilitates direct participation of clinicians in the authoring process.

A PRO*forma* language supports four classes of task: actions, enquiries (that acquire data), decisions and plans. An Action generally represents a request for an external actor to do something (e.g. prescribe or administer a drug, or perform some other clinical intervention). An Enquiry represents a request to an external actor to provide values for data items. A Decision represents a choice between two or more Candidates (beliefs or tasks) in which each Candidate is associated with one or more Arguments. In this model arguments are logical expressions, which if true *support* or *oppose* a candidate. Arguments can be viewed as providing qualitative support or can be given numerical weights where there is a basis for defining such weights.

Two toolsets for creating and executing PRO*forma* models are available. Arezzo is a commercial system available from *Infer*Med Ltd. for Microsoft platforms [4]. Tallis is a java based implementation from Cancer Research UK. This is available for research use and was the tool used in developing TADS.

1.4. TADS Knowledgebase

Due to the wide scope and diverse tasks involved in the Triple Assessment pathway (e.g., genetics, imaging, pathology etc.) no single CPG that has been published in the breast cancer domain was sufficient to cover all the relevant knowledge. In consultation with domain experts, therefore, seven high quality evidence based guidelines were selected as a basis for specifying the logical argumentation for and against each candidate, for each of the four decisions. To determine the quality of each CPG, guideline appraisal instrument AGREE[5] was used. All selected clinical practice guidelines scored high on a 23 point AGREE scale (overall score of > 60% in all domains) suggesting a high level of rigour. Though all the selected guidelines represent high quality sources, only two of them (NICE and NCCN) provided recommendations in clear and easy-to-understand form - the rest of the CPGs were more narrative in form. It therefore required considerable clinical input to transform guideline knowledge into the logical form that TADS could use.

An expert panel was formed which included four senior practicing consultants from four relevant disciplines (surgery, radiology, pathology and genetics). The panel reviewed the knowledge base for its accuracy in encoding the evidence-based guidelines and consensus was achieved. A set of 125 evidence-based arguments derived from the guidelines allowed TADS to exactly replicate the expert panel's recommendations for all decisions on a test set of 15 cases.

[4] http://www.infermed.com/

[5] http://www.agreecollaboration.org/instrument/

2. The TADS Trial

The TADS system was tested in a crossover experiment with a balanced block design in which experienced breast surgeons were asked to consider simulated patients, with and without decision support. A total of 36 practicing clinicians were opportunistically sampled and invited from the population of breast cancer clinicians who conduct triple assessment clinics in the south east of the England. In all, 24 senior breast clinicians agreed to participate.

We constructed fifteen simulated case histories, from a larger number of real cases referred to the Guy's Hospital Triple Assessment clinic over a six month period preceding the study. The fifteen cases were reviewed by the expert panel for the adequacy of data, as well as for internal consistency and representativeness. Three sets of five patients were created, each encapsulating a variety of clinical scenarios.

Each participating clinician was asked to consider two different sets of cases: one set with and one set without decision support: twelve out of twenty four clinicians used the decision support mode first, while the other twelve used the no-decision support mode first, selected randomly. To control for differences in the difficulty of the case sets, the three sets of five cases were also balanced so that each set was included the same number of times in each arm of the trial. We refer to each patient being taken through the four decisions by one clinician as a "patient journey", so in total 120 patient journeys (5 cases times' 24 clinicians) were made with decision support and 120 without.

Trial sessions were conducted using a laptop computer in offices or clinics in the various hospitals in which our participating clinicians worked. All subjects were first familiarised with the system using the same training script. Throughout each session, participants had access to all patient data on paper as well as on TADS. The TADS system captured and kept track of the patient data as well as decision options selected by the user at each decision point, together with a record of what the system's recommendations were in each case.

A semi structured interview was also carried out to collect feedback from clinicians and to perform a questionnaire study. Before going through the trial each participant was asked about his or her general opinion on the usefulness of CDS systems in improving patient care in TA clinic. After using TADS the clinicians were asked about their views on the usefulness of the TADS system in the triple assessment clinic. Both pre-test and post-test opinions were recorded on a five point Likert scale. To investigate changes in the distribution of opinion we used the Fleiss-Everitt simplification of the Stuart-Maxwell test for matched pairs, and then checked for systematic differences using the McNemar test.

The ultimate gold standard to measure the impact of systems like TADS in the real world will be its effect on the patient outcomes, such as cancer detection rate, morbidity or mortality. However the controlled nature of this experiment imposes a less stringent measure or silver standard to assess its impact. The primary outcome measure was the compliance of clinicians with guidelines as judged by our expert panel. After the experiment was completed, the decisions made by each clinician for each patient case were compared to those made by our expert panel. Each "patient journey" was categorised as having had either zero or at least one deviation. The number of patient journeys that contained errors or deviations in both decision support and no decision support arms were compared using a Wilcoxon signed rank test. We had 80% power to find a 15% significant difference (P=0.05).

Deviations were empirically categorised by the expert panel as follows: *Minor or non-critical deviations:* that arguably would not result in direct patient harm but may result in unnecessary utilization of resources, for instance, ordering unnecessary investigation. *Critical deviation:* that could potentially result in patient harm, for instance not performing an essential test recommended by guidelines. A further subgroup of critical deviations was identified as irretrievable critical deviations, where a patient completed the journey and was discharged. In practice, such errors would typically not be spotted or rectified by other members of the team.

3. TADS Trial Results

The primary aim of the study was to evaluate the impact of the TADS system on the clinicians' decision-making performance as judged by their compliance with best practice defined by evidence based guidelines. The number of deviations from the best practice was found significantly lower when TADS provided decision-support to the clinicians. Sixty out of 120 patient journeys undertaken in the no-decision support condition included at least one deviation (see Table 1), compared with only 16/120 when decision support was provided (P < 0.001).

Table 1. Analysis of deviations in decision support and no-decision support arms.

	Without decision support (Total 120 patient journeys)	**With decision support** (Total 120 patient journeys)	
Type of Deviation	Patient journey with at least one deviation of given type	Patient journey with at least one deviation of given type	P value By Fisher's exact
All deviations	60	16	<0.001
Critical deviations	16	1	<0.001
Irretrievable critical deviations	10	1	0.01

The reduction in deviations owing to TADS remained highly significant for all subgroups of deviations including a clinically important subgroup of critical deviations. In the control arm of the trial, 16 critical deviations were identified, compared to only 1 in the decision support arm (P = 0.001). The average time taken by each clinician to complete 10 patient journeys was 37.2 min ranging from 24 to 61 min. There was no significant difference in the mean time taken by clinicians in both groups.

At the end of the trial, we conducted a questionnaire study with participants to learn more about their thoughts on the system. Eleven clinicians became more convinced of its benefit after using TADS; ten clinicians maintained their opinion about the usefulness of such systems and 2 became one category less convinced. There was a highly significant positive shift of the opinion about overall usefulness of the TADS after trialing the system (matched pair, 3 d.f., chi-square=10.26, $P < 0.01$). It should be noted that none of the participants were aware of the trial results at the time of interview.

Common themes that emerged out of the interviews are as follows

- Decision support was most appreciated by clinicians in decisions requiring complex computation, for example the genetic risk assessment decision.
- Most of the clinicians found the transparent way in which guideline recommendations were presented by TADS (allowing the user to verify arguments and drill down into the underlying research evidence if required) very intuitive and helpful so they could follow recommendations with more confidence.
- The strict nature of workflow and task sequencing imposed by TADS was generally viewed as a hindrance in real clinical settings where more flexibility is essential.
- Time was considered as the most precious resource in overburdened healthcare systems and the need to carefully evaluate the impact of systems on consultation times was expressed. Any tool that significantly increases overall time was thought to be virtually non usable in time-pressured environments like NHS hospitals irrespective of its value in improving care.
- Clinicians were more cautious about real time decision support during actual patient encounters and its impact on current doctor-patient interaction. They were more relaxed about having the recommendations prior to patient consultations.
- The lack of an effective mechanism in place for maintenance and updating of the knowledge used by the system was thought by many clinicians to be one of the major reasons for low acceptance of CDS tools by the medical fraternity.
- The need was expressed for smarter data capture interfaces that adapt quickly to changing clinical contexts and patient data, making clinicians' lives easier, as opposed to current clinical information systems which create extra workload through inflexible operation.
- Clinical decisions vary markedly with respect to computational complexity, cognitive load, time pressures and communicative demands. Different forms of decision support may be required for different types of decisions and in certain types of decisions it may not be useful at all.

4. Conclusion

4.1. Strengths and Limitations of the TADS Study

The TADS system has proved to be very effective in improving adherence to the best evidence based practice. A limitation of the trial, however, is that we have used simulated cases rather than real patients so these results need to be treated with caution. Our simulated cases were deliberately more varied than one would find in a typical triple assessment clinic in order to test the effect of the software on differing scenarios. Consequently, they included more difficult cases than one would typically find in 15 cases chosen at random. Simulated case scenarios however have been shown to be good predictors of clinical performance [12].

Furthermore, promising decision support systems may not be deployed in clinical practice due to poor user interfaces and/or poor integration with clinician workflow [13]. The right information must be presented at the right time in the right format without requiring special effort or arrangements [14, 15]. In this study, the strict workflow-based design of the TADS imposed rigidity by compelling the subjects to take all four decisions in the same order, making patient journeys as similar as possible (the only variable being the availability or not of decision support). If we were to trial the software in a real clinical setting, however users are likely to want to enter data and make decisions in any order they wish. PRO*forma* technology allows this but the workflow was not implemented in this way for the trial in order to maintain comparability over conditions.

4.2. Future Directions

Many challenges remain. The TADS trial should be viewed as a proof-of-concept that PRO*forma* and similar task network technologies can support complex clinical workflows and provide guideline-based decision support in a way that is effective and usable. However the key challenge lies in the real world implementation of the system. The list of issues raised by the participants in the trial highlights some of the important problems that the decision support community should attend to before we may realistically expect widespread acceptance of such systems. The CREDO project aims to address these questions over the next few years.

Acknowledgements

Thanks to Chris Hurt, Sharon Love, Arnie Puroshottham, Rory Steel, Matthew Williams, Alyssa Alabassi, Victoria Monaghan, and Mike Bradburn for their input into the inception and conduction of this project.

References

[1] C. B. Granger, P. G. Steg, E. Peterson, J. Lopez-Sendon, F. Van de Werf, E. Kline-Rogers, J. Allegrone, O. H. Dabbous, W. Klein, K. A. Fox and K. A. Eagle, "Medication performance measures and mortality following acute coronary syndromes," Am J Med 118(8), 858-865 (2005).

[2] E. D. Peterson, M. T. Roe, J. Mulgund, E. R. DeLong, B. L. Lytle, R. G. Brindis, S. C. Smith, Jr., C. V. Pollack, Jr., L. K. Newby, R. A. Harrington, W. B. Gibler and E. M. Ohman, "Association between hospital process performance and outcomes among patients with acute coronary syndromes," JAMA 295(16), 1912-1920 (2006).

[3] J. Fox, N. Johns, C. Lyons, A. Rahmanzadeh, R. Thomson and P. Wilson, "PROforma: a general technology for clinical decision support systems," Comput Methods Programs Biomed 54(1-2), 59-67 (1997).

[4] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. H. Shortliffe and M. Stefanelli, "Comparing computer-interpretable guideline models: a case-study approach," J Am Med Inform Assoc 10(1), 52-68 (2003).

[5] R. T. Walton, C. Gierl, P. Yudkin, H. Mistry, M. P. Vessey and J. Fox, "Evaluation of computer support for prescribing (CAPSULE) using simulated cases," BMJ 315(7111), 791-795 (1997).

[6] C. Tural, L. Ruiz, C. Holtzer, J. Schapiro, P. Viciana, J. Gonzalez, P. Domingo, C. Boucher, C. Rey-Joly and B. Clotet, "Clinical utility of HIV-1 genotyping and expert advice: the Havana trial," AIDS 16(2), 209-218 (2002).

[7] P. Taylor, J. Fox and A. T. Pokropek, "The development and evaluation of CADMIUM: a prototype system to assist in the interpretation of mammograms," Med Image Anal 3(4), 321-337 (1999).

[8] J. Bury, C. Hurt, A. Roy, L. Cheesman, M. Bradburn, S. Cross, J. Fox and V. Saha, "LISA: a web-based decision-support system for trial management of childhood acute lymphoblastic leukaemia," Br J Haematol 129(6), 746-754 (2005).

[9] A. S. Coulson, D. W. Glasspool, J. Fox and J. Emery, "RAGs: A novel approach to computerized genetic risk assessment and decision support from pedigrees," Methods Inf Med 40(4), 315-322 (2001).

[10] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde and W. Bob. Knowledge Engineering and Management: The CommonKADS Methodology, 1st edn, The MIT Press. (2000).

[11] D. R. Sutton and J. Fox, "The syntax and semantics of the PROforma guideline modeling language," J Am Med Inform Assoc 10(5), 433-443 (2003).

[12] J. J. O'Hagan, L. J. Davies and R. K. Pears, "The use of simulated patients in the assessment of actual clinical performance in general practice," N Z Med J 99(815), 948-951 (1986).

[13] M. Eccles, E. McColl, N. Steen, N. Rousseau, J. Grimshaw, D. Parkin and I. Purves, "Effect of computerised evidence based guidelines on management of asthma and angina in adults in primary care: cluster randomised controlled trial," BMJ 325(7370), 941 (2002).

[14] D. W. Bates, G. J. Kuperman, S. Wang, T. Gandhi, A. Kittler, L. Volk, C. Spurr, R. Khorasani, M. Tanasijevic and B. Middleton, "Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality," J Am Med Inform Assoc 10(6), 523-530 (2003).

[15] B. C. James, "Making it easy to do it right," N Engl J Med 345(13), 991-993 (2001).

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-243

Lessons Learned from Adapting a Generic Narrative Diabetic-Foot Guideline to an Institutional Decision-Support System

Mor PELEG[a],[1], Dongwen WANG[b], Adriana FODOR[c, d], Sagi KEREN[e] and Eddy KARNIELI[c, d]

[a]*Department of Management Information Systems, University of Haifa, Israel*
[b]*Biomedical Informatics Program, University of Rochester, NY*
[c]*Institute of Endocrinology, Diabetes & Metabolism, Rambam Medical Center and*
[d]*Galil Center, Technion–Israel Institute of Technology, Israel*
[e]*Department of Computer Science, University of Haifa, Israel*

Abstract. Clinical guidelines usually need to be adapted to fit local practice before they can be actually used by clinicians. Reasons for adaptation include variations of institution setting such as type of practice and location, availability of resources, differences in patient populations, local policies, and practice patterns. When a guideline is implemented for clinical decision support and integrated with an institution's clinical information system, the data model of the local electronic medical record (EMR) and the data actually collected and stored in it also influence the guideline's adaptation. The purpose of this work is: (1) to characterize a tool-supported process for guideline encoding that addresses local adaptation and EMR integration, and (2) to identify the types of changes in guideline encoding during the local adaptation process.

Keywords: Clinical guidelines, computer-interpretable guidelines, local adaptation, EMR integration, GLIF3

Introduction

Clinical guidelines are systematically-developed statements to assist practitioner and patient decision making about appropriate healthcare for specific clinical circumstances [1]. They aim to improve quality of care, reduce unjustified practice variations, and control costs. Studies have shown that computerizing guidelines as decision-support systems (DSSs) is more likely to generate positive impacts when compared to paper-based guidelines [2]. Therefore, many researchers have been developing languages to represent guidelines in computer-interpretable formats [3].

Clinical guidelines developed by government agencies or medical specialty organizations at the national level usually need to be adapted to fit the local practice

[1] Corresponding Author: Mor Perleg, Department of Management Information Systems, University of Haifa, Haifa, Israel, 31905, E-mail: morpeleg@mis.hevra.haifa.ac.il

before they can be actually used by clinicians at specific institutions. Reasons for such local adaptations include variations of institution setting, such as type of practice (e.g., hospital vs. physician office) and location (e.g., urban vs. rural), availability of resources, difference between patient populations (e.g., as reflected in prevalence of a disease), local policies, and practice patterns [4, 5]. Successful guideline implementation has to be supported by organizational efforts [6]. At the same time, there are many technical issues that should be addressed, especially when implementing guidelines as clinical decision support systems (CDSSs).

Several groups have reported their experiences when adapting narrative clinical guidelines and implementing them as CDSSs. Shiffman investigated the validity of guideline knowledge and suggested methods to assess the logic integrity of the original guideline, such as completeness and consistency of decision recommendations. An exhaustive set of possible values for each variable should be determined and special attentions should be devoted to the logic combination of variables within decision criteria [7]. HieroGLIF [8] and CAMINO [9] are approaches that separate the site-independent information of guidelines from site-specific information. Based on the theory of axiomatic design (AD), HieroGLIF is an extension of the GLIF3 [10] guideline modeling language, supporting hierarchical modular guidelines, starting from broad objectives and ending with details of the recommendations. Guidelines are specified in a setting-independent manner and are then locally adapted by each practice. Changes occur mostly at the lower, more detailed levels of the design tree. CAMINO is a tool that allows formalizing site-specific customizations, including: (1) annotations documenting the guideline authors' assumptions, (2) a model of organization activities, resources, policies, and preferences, and (3) rules that define allowed transformations that make a generic guideline site-specific. GLARE [11] takes a different approach to adaptation of a guideline. The resources needed by each action of the guideline are explicitly declared. Then, a context-specific guideline is built from the general one by collecting only those paths that require available resources. Tierney et al. investigated the practical considerations when adapting an inpatient heart failure guideline to the outpatient setting and implementing it within an electronic medical record (EMR) [12]. About one-third of the original guideline’s recommendations were not included in the new implementation because of the differences in the setting. In addition, they had to translate some of the guideline’s data definition into multiple EMR entries as the original definitions of these data in the guideline were not directly available from the local EMR.

These works referred to two types of processes: (1) adaptation needed in order to formalize a textual guideline, including knowledge interpretation and validation of the completeness and correctness of guideline knowledge, and (2) adaptations needed to fit the context (including local procedures, resources, and EMR). In this paper, we address mainly the second kind of adaptation, to which we refer as *local adaptation*. Our study has two specific purposes: (1) to identify and classify specific types of changes in guideline encoding during the process of local adaptation of a guideline for clinical decision support, and (2) to suggest a process for local adaptation that addresses the issue of data availability, clinical workflow, and validation of encoding. These are important issues that need to be addressed whenever a guideline encoding is shared, i.e., implemented and integrated with EMR systems at a local institution.

1. Methods

To adapt a generic guideline in narrative format as a CDSS at a local institution, we assumed: (1) clinicians' needs of decision/management support services from a computer system have already been identified, and (2) a paper guideline has already been chosen as the source of medical knowledge such that it can be adapted for implementation. We used the GLIF3 [10] guideline modelling language to encode guidelines. A GLIF3 encoding of a guideline includes a visual clinical algorithm that specifies a process of care, depicting patient states, medical tasks, and clinical decisions, all embedded within a specific workflow. The medical tasks, such as prescriptions of drugs and orders of lab tests are specified by concepts taken from controlled vocabularies. The decision criteria are specified through logic expressions that reference clinical concepts, which are then mapped to EMR data. We chose Protégé-2000 [13] as our vehicle for guideline encoding in GLIF3.

We analyzed the encoding of a diabetes foot care guideline developed by the American College of Foot and Ankle Surgeons [14], which was selected by our clinician experts for implementation. Our analyses focused on changes that were made throughout the process of translation of the original guideline into the GLIF3 format as well as its integration with a local EMR. The front-end of the EMR is a web-based patient clinical record, developed by one of the authors (EK), which is used by clinicians to manage diabetic patients, from local as well as remote clinics. The system uses an internet browser as a user-friendly interface and an Oracle database server located at the Rambam Medical Center is the backbone of the system. A built-in internal email application serves for communication between the clinicians and consultants while reviewing of the data. Our final goal was to integrate guideline recommendations with this system.

The process to encode and to adapt the narrative guideline consisted of:

Step 1 – translation of the original guideline into GLIF3 format. One of the informaticians on the team (MP) first performed a high-level encoding in GLIF, where decision criteria were written in English, rather than using formal notation. The encoding was based only on the original narrative guideline without consideration of local adaptation. The rationale of doing this was to facilitate communication with clinicians, since the initial conceptual flowchart created at this step could show patient management process, and thus the clinicians would understand better how the guideline could fit with their usual workflow. At this stage, we only specified the initial version of the conceptual algorithm with a high-level description of the action steps and the decision steps written in English rather than in GLIF3's formal definition.

Step 2 – analyses of local practice. After the initial encoding was completed, the two expert clinicians in our team (EK, AF), together with an informatician (MP), clarified the definitions of concepts, matched the data items of the initial encoding to the data entities in the local EMR, and analyzed the fitness of the guideline to the local setting and workflow.

Step 3 – revision of the initial encoding. Based on the analyses in the previous step, we made revisions to the initial encoding such that: (1) it fit with local practice, and (2) it included formal definitions of data queries and decision criteria that were mapped to the EMR's schema and available data.

Step 4 – validation: manual check of encoding. The clinicians and informatician manually checked the clinical algorithm, formal definitions of decision criteria, and mapping of concepts to the EMR schema.

Step 5 – validation: execution of test cases by informaticians. For further validation, MP and DW used the GLIF Execution Engine (GLEE) [15] to apply the encoded guideline to test cases. For this purpose, SK integrated GLEE with the back-end Oracle database of the local EMR, so that GLEE can directly query the patient data of the test cases. We invoked the guideline in an interactive mode, following the path of execution when it was applied to a specific case. For the purpose of validation, we used 14 real cases and 6 simulated cases. The simulated cases supplemented the real cases by testing branches of the clinical algorithm that were not tested by the real cases. The Human Rights Committee of Rambam Medical Center approved the study, and written informed consents were obtained prior to adding patient data to the EMR. Figure 1 shows a screenshot of the guideline being executed by GLEE.

It is important to note that the validation (steps 4 and 5) was an iterative process. When problems were found in steps 4 and 5, we went back and repeated steps 2 and 3.

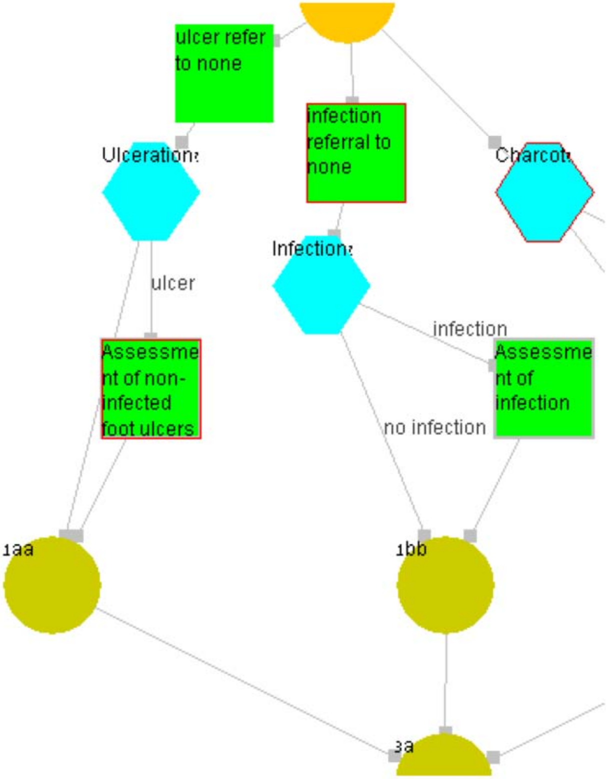


Figure 1. A screenshot of the diabetic foot guideline being executed by GLEE. Squares indicate Action Steps, hexagons – decision steps, circle with 'p' on it – Branch Step, and blank circles – Synchronization Steps. GLEE enables parallel execution. The branch step at the top of the figure enables parallel execution of three parallel paths dealing with different foot problems: ulcer (left path), infection (middle path), and Charcot (malalignments of the foot bones – right path). The two action steps "ulcer refer to none" and "Infection referral to none" set flag variables that guard that referrals, due to ulcer or due to infection, are made only once during the algorithm. At anytime, one of the currently active steps can be chosen to execute. In the figure, the three steps with bold contour are active, each belonging to a different path of the three paths explained above. The one with the heavy black contour, labelled "Assessment of non-infected Foot ulcers", is chosen to be executed, leading to entering a subguideline. GLEE evaluates decision criteria, based on patient data stored in the EMR, and automatically suggests which branch to follow. In the figure, the 'Ulceration?' Decision was evaluated and the 'Ulcer' path has been followed.

Step 6 – validation: execution of test cases by expert physicians. A year after the guideline was finalized, we decided to develop a new user-interface for the system. In preparation, we carried out another round of validation and updates that considered different types of patient-clinician encounters: those that constitute the main care algorithm and those that are follow-up sessions (follow-up on antibiotic treatment).

2. Results

During the iterative encoding process, we found that we had to clarify a few concepts that were important for decisions. In addition, we had to change the guideline encoding in different ways to fit with the local setting, practice, and workflow. Here, fitting with local setting concerns the availability of resources that depends on the type of institution and its location, fitting with local practice concerns the different treatment selections that depend on local policy or patient population, and fitting with workflow concerns the delivery of guideline recommendations that could be seamlessly integrated with clinicians' practice. When integrating the encoded guideline with the local EMR, we found that we had to further revise the guideline's encoding to map to the unique data model at the local site and to address the issue of data unavailability (never collected by clinicians even though they were modelled in the EMR schema).
These revisions were significant, affecting the conceptual flowchart, the definition of data items, and the formal specification of decision criteria.

2.1. Clarifications of Concepts

The original guideline contained implicit background knowledge. For example, the definitions of a few high-level concepts (e.g., ischemic symptoms) were not explicitly given. In addition, several data items in the original guideline were not fully specified (e.g., the data type and expected value of ulcer margins). Specifically, the original guideline contained ten concepts that were used at decision points of the guideline (e.g., neuropathy or not). Two of them (neuropathy, inflammation) were not defined explicitly and our expert clinicians had to provide clarifications for them so that they could be mapped to EMR data items. One concept definition had to be changed to fit with the local practice (PVD), see Figure 2(a). The definitions of six other concepts (charcot, pulses present, normal results for non-invasive tests, inflammation, limb-threatening infection, infection) were restated according to the EMR schema and available data (Figure 2(b)), while conserving the meaning of the original guideline. In the validation phase, the definition of neuropathy was changed again because the clinicians were not satisfied with their previous definition.

2.2. Fitting with Local Setting

The setting of outpatient clinics in which the guideline is being implemented in Israel is different from that assumed by the American guideline. For example, in the Israeli setting, family practitioners refer patients to orthopaedic or plastic surgeons for wound debridment (i.e., surgical removal of dead tissue from a wound); while in the US, this is performed at the local clinics. Thus at step 2, we made three changes to guideline encoding to fit with our setting: (1) patients who have charcot malalignments should be referred to a hospital or diabetic foot clinic, instead of being managed by primary practitioners as suggested by the original guideline, (2) patients with severe ulcers should be referred to hospitals for treatment, instead of having their wounds debrided by the primary practitioner, and (3) primary practitioners cannot give parenteral antibiotics to patients with non limb-threatening infection, but only oral antibiotics, which are easily administered by the patients themselves at their own homes. At the validation step, we made another change of this type: transcutaneous oximetry should be ordered only if the patient is referred to be hospitalized.

(a) Change to fit with local practice

Initial encoding:
pulse_popliteal = false or
pulse_femoral = false or
pulse_dorsalis_pedis = false or
pulse_posterior_tibial = false or
cyanosis = true

Final encoding:
// absent popliteal or femoral pulses of either foot
pulse_popliteal_L_absent = "Yes" or pulse_popliteal_R_absent = "Yes" or
pulse_femoral_L_absent = "Yes" or pulse_femoral_R_absent = "Yes" or
// blue skin in addition to a diminished pulse or missing posterior-tibial or
// dorsalis_pedis pulse
(skin_color_blue_purple = "Yes" and
(pulse_popliteal_L_diminished = "Yes" or pulse_popliteal_R_diminished = "Yes" or
pulse_femoral_L_diminished = "Yes" or pulse_femoral_R_diminished = "Yes" or
pulse_dorsalis_pedis_L_absent = "Yes" or pulse_dorsalis_pedis_R_absent = "Yes" or
pulse_dorsalis_pedis_L_diminished = "Yes" or pulse_dorsalis_pedis_R_diminished = "Yes" or
pulse_posterior_tibial_L_absent = "Yes" or pulse_posterior_tibial_R_absent = "Yes" or
pulse_posterior_tibial_L_diminished = "Yes" or pulse_posterior_tibial_L_diminished = "Yes")) …

(b) Change to map to EMR schema
Initial encoding: charcot = true
Final encoding: (mal_forefoot_L = "Yes" or mal_forefoot_R= "Yes" or mal_mid_L = "Yes" or mal_mid_R = "Yes" or mal_hind_L = "Yes" or mal_hind_R = "Yes") and redness = "Yes" and swelling = "Yes"

Figure 2. The initial and final encoding of decision criteria. (a) decision criterion for peripheral vascular disease (PVD) was changed to fit with the local practice. According to the original guideline (initial encoding), four pulses were measured in each leg, and if at least one of them was absent, or if the foot colour was blue (cyanosis), then PVD was concluded. The local practice (final encoding) is that the physician documents whether a pulse is missing or diminished. If either the popliteal or the femoral pulse is absent in either the right (R) or left (L) foot, it is sufficient to conclude PVD; but if these pulses are diminished, or if either the dorsalis pedis or the posterior tibial pulse is absent or diminished, the foot colour should be blue in order to conclude PVD. (b) The initial decision criterion for charcot was changed to map to the EMR schema. In this case, the information on whether a patient has charcot is dispersed over many EMR fields that designate areas of malformations of the foot: forefoot, mid-foot, hind-foot. To conclude Charcot, signs of redness and swelling have to be present in addition to malformations.

2.3. Integration with Workflow

We made one change to the guideline encoding in step 2 to integrate with clinical workflow. Our clinicians noted that although the guideline defined criteria to categorize ulcer grade, this function was not required in the implementation because determination of ulcer grade was performed at EMR data entry after physical exam. At the validation step (step 4), we made three additional changes: (1) two courses of antibiotics (instead of one in the original guideline) should be tried to treat infection before referring the patient to a hospital, (2) telemedicine consultation was added to determine ulcer staging, and (3) the clinical algorithm of the encoded guideline was rearranged so that all of the referral actions were put into one place of the top-level guideline to fit with the referral task in practice. At validation step 6, we introduced two changes relating to workflow integration: (1) the design of the encoded guideline was changed to reflect different encounter types, and (2) adding decision-support for determining ulcer grade that could be used if the grade was not determined during data entry. Note that this change overrode the second category of changes made at step 4, as described earlier.

2.4. Matching with Local Practice

We made three changes to match with the local practice: (1) the decision criteria for suspecting PVD were changed (Figure 2a), (2) several medical actions were added (e.g., ordering EMG) or removed (e.g., clonus testing), and (3) a physician who encounters a patient with non- life-threatening infection may treat him with antibiotics as suggested by the guideline, or may decide to refer the patient to a hospital or clinic.

2.5. Consideration of Data Retrieval

To ensure that the guideline can be directly applied to specific patient cases, we studied the EMR schema and the available data. We identified 188 data items of the encoded guideline that should be mapped to the data fields of the local EMR. The EMR schema and the availability of data actually collected and stored in the EMR affected the encoding of decision criteria in the following ways.

- Multiple guideline concepts could be mapped to a single EMR data item. For example, both abscess and fluctuance were mapped to abscess.
- A single guideline concept could be mapped to combinations of multiple EMR data items. For example, charcot was mapped to redness, swelling, as well as one of the six different data items that describe malformations in different locations of the right or left feet (see Figure 2(b)).
- Guideline concepts missing in the EMR schema. The expert clinicians decided whether to remove the unavailable data items from the definitions of decision criteria or to change the EMR schema such that the relevant data can be collected.
- Data unavailability despite the fact that the EMR schema allowed easy mapping. For example, while there was an EMR field to record the existence of ulcers, the data were never actually collected in our EMR.
- Mismatch of data type and definition of normal ranges. For example, the guideline refers to a test result that needs to be greater than 1.3, whereas the EMR stores the values as three possible strings: greater_than_1.1, less_than_0.5, between 0.5_0.9.

2.6. Changes of Guideline Encoding

The local adaptation process had significant affects on the encoding, including change of algorithm design, definition of decision criteria, and specification of data items that are referenced by the decision criteria. Table 1 shows the number of knowledge components created during the iterative encoding process. The number of components that define the structure of the algorithm – action and decision steps – decreased significantly during the local adaptation step, resulting in version 1. This was mainly due to the removal of the charcot branch of the algorithm to fit with our setting. Although the direction of the change (reduction) is only specific to this guideline, we believe that in general, the most significant scale of changes in the structure of the clinical algorithm will occur during this stage. The number of decision steps increased during the validation step, mostly due to the change of the algorithm's design to group all decisions regarding referrals to a single place in the top-level guideline.

Table 1. The number of guideline components and changes made at different steps of encoding.

	Versions				
Knowledge Item	Original	V1	V2	V3	V4
Decision steps	23	13	13	21	23
Action steps	84	60	60	60	73
Decision criteria	9	52	35	50	56
Data items	15	73	66	150	157

Original – original encoded version; V1– version 1 created after locally adapting the guideline with the clinicians (steps 2 and 3); V2 - version 2 created after the initial clinical validation (step 4 and 5); V3 – semi-final version created after iterative validations; V4 – final version created at validation step 6

The components that define the computable specification include decision criteria and data items. During the creation of the original encoding, the informatician wrote the majority of the decision criteria in English. According to our local adaptation process, formal definitions are created during local adaptation, and are changed during validation, as reflected in Table 1. The number of data items increased during validation, in which the mapping of guideline concepts to EMR terms was determined, often involving mapping a single guideline concept to multiple EMR terms.

We validated the guideline using GLEE by executing 14 real patient cases from the EMR, supplemented by six simulated cases to cover all paths through the algorithm. The validation considered 22 branching points and recommendations. At the end of the validation, all 22 criteria matched with the expected results.

During a later phase of the development of our guideline-based DSS, which addressed a different user interface for the system, the guideline was revised (as described in section 2.3). We carried out a validation study (paper in preparation) in which eight clinicians evaluated the correctness of the encoding based on the recommendations given by the CIG to a simple and a complex patient scenario. This evaluation resulted in a change to one of the action steps.

3. Discussion

As manifested by the works covered in this book, the community of researchers in the field of computerized guidelines has been recently focusing on the following issues

- Achieving high-quality and safe guidelines by using ontologies for forma guideline representation (Dominguez et al., [10]) and formal methods to verify guideline encodings (Terenziani et al., Chesani et al., Hommersom et al., Balser et al.)

- Shortening the life-cycle of implementation of guidelines as DSSs, by:
  - Gradually marking up narrative guidelines and transforming them into computer-interpretable representations (Hatsek et al.)
  - Supporting local adaptation of guidelines and their integration with local systems (Hommersom et al., Patkar et al., [4])
  - Allowing for sharing of CIGs by multiple implementing institutions [10]
  - Developing tools to support the guideline life cycle, such as guideline editing, verification (see above), and execution (Votruba et al.)

Our work fits within these current research trends; our thorough encoding and verification process was carried out in order to achieve a high quality CIG, encoded in the GLIF3 ontology, suitable for the local environment, and integrated with the EMR.

We proposed a process for translating a narrative guideline and adapting it as a CDSS in a local institution. This process used tools to support encoding and validation. We focused on changes made during a local adaptation process. This is in contrast to other studies that focused on the changes during the development of the original clinical algorithm without considering a computable representation [16] or the maintenance of a guideline to reflect the latest development in medical knowledge [17].

The process that we carried out involved initial encoding of the original guideline by informaticians. This approach has benefits and drawbacks. Starting the encoding from the original guideline makes it potentially possible to adapt it for use by several local institutions with modifications to specific parts of the general encoding. Indeed, the lesson we learned is that a significant portion of the original guideline is useful for the local site. This approach also makes it easier to communicate with clinicians to demonstrate the system to them, to identify the potential problems of the original guideline, and to find appropriate solutions to a local implementation. However, if we only care about implementation at a specific site, changing the process by having informaticians and clinicians working together as early as possible would save time, especially when significant parts of the original guideline need to be changed. A final lesson we have learned about the development process is that we should consider integration with EMR as early as possible, as it also has major impacts to the encoding.

One of the goals when developing the GLIF3 model was to allow sharing of encoded guidelines among institutions and platforms. Changes of guideline encoding due to differences in local setting, practice, and workflow seem to be inevitable. Further investigations are required to study whether it is possible and, if it is, how to separate the medical knowledge from the organizational knowledge when developing guidelines from clinical decision support. Obviously, the encoding of data items should not include proprietary terms or codes belonging to some institution's EMR, but rather should refer to a virtual medical record that defines a high-level structure of clinical data based on controlled vocabularies (e.g., SNOMED-CT [18]). Because right now there is little consensus on such a virtual medical record, our implementation was based on a 1:1 mapping between concept identifiers and fields of EMR tables, which is a limitation. Work in progress includes a mapping ontology that would allow encoding the guideline in GLIF through clinical abstractions and mapping to the actual EMR tables. This mapping ontology would support the definition of temporal (e.g., current, recent) and taxonomical (e.g., antibiotics) abstractions that would simplify guideline encoding in GLIF without the need to change the GLIF language.

Our work enabled us to test the potential ways in which guideline-based interventions could be integrated with the clinician's workflow. Our ongoing work involves the integration of the decision support functions within the web-based user interface that the clinicians currently use to enter data into the EMR. Our long-term goal is to evaluate the impact of the system to clinical process and outcome, such as documentation of patient data, order of lab tests, and patient performance.

Acknowledgements

We thank Mr. A. Kulater for his excellent technical assistance. EK was supported by grants from Israel National Institute for Health Policy and Health Services Research and Galil Center for Telemedicine and Medical Informatics. We thank the anonymous reviewers for their suggestions.

References

[1] Field MJ, Lohr KN. Guidelines for Clinical Practice: Directions for a New Program. Washington DC: Institute of Medicine, National Academy Press; 1990.

[2] Shea S, DuMouchel W, Bahamonde L. A Meta-analysis of 16 Randomized Controlled Trials to Evaluate Computer-based Clinical Reminder Systems for Preventative Care in the Ambulatory Setting. J Am Med Inform Assoc 1996;3(6):399-409.

[3] Peleg M, Tu SW, Bury J, Ciccarese P, Fox J, Greenes RA, et al. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. J Am Med Inform Assoc 2003;10(1):52-68.

[4] Boxwala AA, Tu S, Zeng Q, Peleg M, Ogunyemi O, Greenes RA, et al. Towards a representation format for sharable clinical guidelines. J Biomed Inform 2001;34:157-69.

[5] Rhinehart E, Goldmann DA, O'Rourke EJ. Adaptation of the Centers for Disease Control guidelines for the prevention of nosocomial infection in a pediatric intensive care unit in Jakarta, Indonesia. Am J Med 1991;16(91(3B)):213S-220S.

[6] Nonino F, Liberati A. Essential requirements for practice guidelines at national and local levels. Neurol Sci 2004;25:2-7.

[7] Shiffman RN. Towards effective implementation of a pediatric asthma guideline: integration of decision support and clinical workflow support. Proc Annu Symp Comput Appl Med Care; 1994:797-801.

[8] Scott-Wright AO, Fischer RP, Denekamp Y, Boxwala AA. A methodology for modular representation of guidelines. Medinfo; 2004:149-53.

[9] Fridsma DB, Gennari JH, Musen MA. Making generic guidelines site-specific. Proc AMIA Symp; 1996:597-601.

[10] Boxwala AA, Peleg M, Tu S, Ogunyemi O, Zeng Q, Wang D, et al. GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. J Biomed Inform 2004;37(3):147-61.

[11] Terenziani P, Montani S, Bottrighi A, Torchio M, Molino G, Correndo G. A context-adaptable approach to clinical guidelines. Medinfo; 2004:169-73.

[12] Tierney WM, Overhage JM, Takesue BY, Harris LE, Murray MD, Varg DL, et al. Computerizing guidelines to improve care and patient outcomes: the example of heart failure. J Am Med Inform Assoc 1995;2(5):316-22.

[13] Gennari J, Musen MA, Fergerson RW, Grosso WE, Crubezy M, Eriksson H, et al. The Evolution of Protege: An Environment for Knowledge-Based Systems Development. Intl J Human-Comp Interaction 2002;58(1):89-123.

[14] American College of Foot and Ankle Surgeons. Diabetic foot disorders: a clinical practice guideline. J Foot Ankle Surg 2000;39(5 Suppl):S1-60.

[15] Wang D, Peleg M, Tu SW, Boxwala AA, Ogunyemi O, Zeng Q, et al. Design and implementation of the GLIF3 guideline execution engine. J Biomed Inform 2004;37(5):305-18.

[16] Peleg M, Patel VL, Snow V, Tu S, Mottur-Pilson C, Shortliffe EH, et al. Support for Guideline Development through Error Classification and Constraint Checking. Proc AMIA Symp. 2002:607-11.

[17] Peleg M, Kantor R. Approaches for guideline versioning using GLIF. Proc AMIA Symp. 2003:509-13.

[18] SNOMED International. SNOMED Clinical Terms. 2006. http://www.snomed.org/snomedct/index.html

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-253

Verification of Medical Guidelines in KIV

Jonathan SCHMITT [1], Michael BALSER and Wolfgang REIF
University of Augsburg, Germany

Abstract. We propose to use computerised medical guidelines as models for verification tools, so they can be validated with medical properties. To test the applicability we provide an implementation of the semantics of the medical planning language Asbru and also provide a formalised guideline for the treatment of breast cancer. With this case study we conduct experiments testing different proof techniques to cope with several challenges which guidelines provide.

Keywords. Interactive verification, Asbru, Protocure, KIV

Introduction

Guideline development groups spend significant effort on improving the quality of their guidelines. One of the trends in the recent past is the shift towards evidence based medicine. Also, more and more guidelines are evaluated using tools like the AGREE instrument, improving the quality and the transparency of the guideline development process. On the other hand, medical guidelines typically are informal documents. It is an old truth, that informal texts may contain inconsistencies and may be incomplete. Such errors are intolerable in a medical guideline, which may affect the welfare of hundreds to thousands of patients.

We therefore propose the use of the most extreme methods of quality assurance known in the domain of software engineering in the medical field. We research how these formal methods fit in this non-standard application area and how it is possible to interpret medical guidelines as 'computer programs' for the tools and techniques to be applicable.

In a first step, we model the guideline in a formal language. Already, the modelling process points to errors in the informal guideline. In a second step, we formally verify properties of the guideline to validate the formal model and to further improve the quality of the original guideline. Formal verification of properties requires sophisticated tool support in practice.

In this chapter, we present the first tool which supports interactive verification of medical guidelines. We have added proof support for guidelines modelled in the Asbru [8] language to the KIV theorem prover [4]. *Challenges which have been solved are as follows.* (I) Instead of encoding Asbru medical guidelines in a different formalism already supported in the theorem prover, we have defined the Asbru semantics in KIV. Asbru plans translate almost one-to-one to the interactive theorem prover. (II) The intuitive strategy of symbolic execution as proof method has been carried over to reasoning

[1] Corresponding Author: Jonathan Schmitt, University of Augsburg, Lst. Reif, 86135 Augsburg, Germany; E-mail: schmitt@informatik.uni-augsburg.de

in Asbru. As a consequence, proofs are straightforward and can be automated to a large extend. This is important for verification in practice.

A number of languages apart from Asbru exist to model medical guidelines, for example EON [9], GLIF [10], GUIDE [12], PRODIGY [11], and PROforma [6]. We have chosen Asbru, because Asbru allows for complex timing conditions and because a formal semantics has been defined [1]. As an alternative to interactive verification, automatic techniques, especially model checking, can be used to analyse medical guidelines. We have already utilised model checking for the verification of guidelines in [5]. Interactive verification complements the use of model checking to also verify large and complex medical guidelines.

1. The Protocure Approach

The work presented in this chapter is the final quality assurance step in a large process aiming at the improvement of medical guidelines. The process starts out at the informal medical guidelines and protocols, which are used today. These guidelines are seen as informal models of the treatment or diagnosis of a disease and then stepwise transformed into a formal model which uses the Asbru language. This formalisation process makes use of sophisticated AI techniques like pattern recognition [15] to reduce the effort necessary.

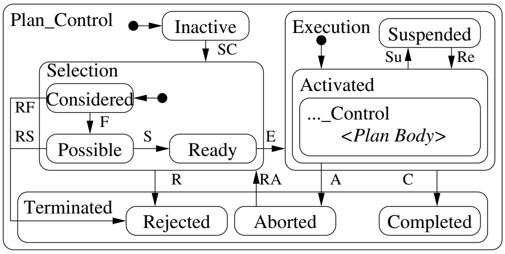
The formalisation has been found to be a very complex and difficult task, which is why an intermediate language called MHB [17] has been defined and used, which can be seen as a link between the informal document and the formal Asbru model. The formalisation process and the benefits of using such an intermediate language have been documented [16].

Parallel to the task of formalising the guideline, properties have to be identified and formalised to assess the quality of the guideline [19]. These indicators may be derived from other guidelines treating the same disease, from medical authorities or from medical background knowledge.

Once guidelines and properties have been formalised, both can be put together to verify that the formalised guideline adheres to the properties, which is the final quality assurance step described within this chapter. However, it should be noted, that even the formalisation process itself may reveal omissions or inconsistencies within the guideline.

A current trend in guideline development is the introduction of so called living guidelines. This is a concept where a guideline is no longer developed from scratch every four to six years. Instead the guideline is updated every half year. This should be reflected in the quality assurance methods, such that much of the effort spent on the validation of a guideline is saved, once an update occurs. This has also been done within the Protocure context and has been documented in a Protocure II project deliverable [2].

[2] http://www.keg.uji.es/deliverables/annex-D6-Living-Proofs.pdf



F : satisfied(filter) and Activated(parent)
RF : not satisfiable(filter)
S : satisfied(setup) and Activated(parent)
RS : not satisfiable(setup)
R : Terminated(parent)
Su : satisfied(suspend)
Re : satisfied(resume)
C : satisfied(complete)
A : satisfied(abort) or Terminated(parent)

Figure 1. Plan state model of a single Asbru plan

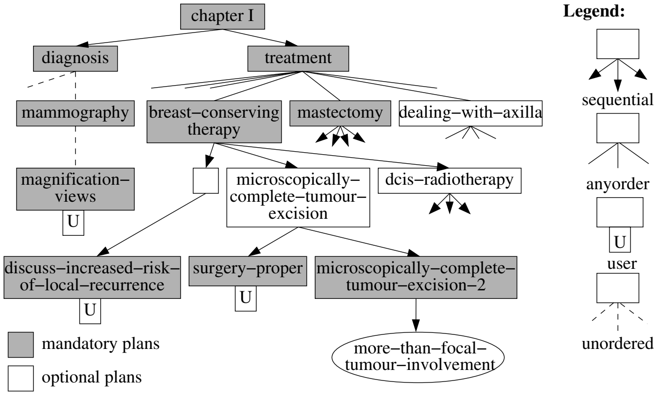


Figure 2. Part of chapter 1 of 2002 version of Dutch breast cancer guideline

2. Medical Guidelines in KIV

2.1. Asbru modelling language

Asbru [8] is a plan oriented language. Plans are organised in a hierarchy of plans. A plan can refer to a number of sub plans in its plan body. Plans without sub-plans represent medical interactions, while Plans with sub-plans describe the organisation of treatment. Plan execution of a single Asbru plan adheres to a so called plan state model, which is visualised in Figure 1. Filter and setup conditions (F and S) are used to control the applicability of a plan, abort and complete conditions (A and C) are used to monitor execution. As a speciality of Asbru, conditions can be monitored over time according to so called time annotations. The sub plans in the plan body can be organised using different body types (e.g., sequential, any-order, or parallel). The current state of a plan – especially if a plan has been rejected, aborted, or completed – is propagated according to the plan hierarchy to its super and sub plans. If a plan is mandatory, it must be completed for its super-plan to complete in return. If the plan is not mandatory, it may also be rejected or aborted in certain cases and still allow the super-plan to complete. The semantics of Asbru is formally defined in [1]. The semantics of the Asbru timing aspects has been defined in [14].

```
    asbru(bct)
  = mk-asbru-def(
      /* filter condition with default time annotation */
      mk-aasbruc(
      /* plan is only applicable if patient prefers bct */
        mk-acond(λ pd. pd.patient-prefers-bct, default-ata), false, false),
      /* other conditions as default conditions */
      setup_condition, suspend_condition, resume_condition,
      abort_condition, complete_condition,
      /* sequential body, aborted sub plans are not retried */
      sequential,
      /* do not restart aborted sub-plans */
      false,
      /* list of sub plans */
      bct1 + mcte + dcis + [],
      /* sub plans are optional, body waits for optional */
      wait-for-n(0, bct1 + mcte + dcis + []),
      /* do not wait for optional sub-plans if otherwise finished */
      false);
```

Figure 3. Example Asbru plan in KIV

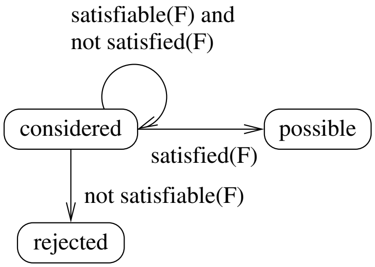
Figure 2 shows part of the Asbru model of the (outdated) 2002 version of a Dutch breast cancer guideline. The top-level Asbru plan sequentially executes diagnosis and treatment. The different treatment options, breast-conserving therapy, mastectomy, treatment of the axilla, and others, are executed in sequentially but not in a fixed sequence. The different options are guarded by filter conditions (not shown in Figure 2). The treatments are further refined by their sub-plans. Overall, the Asbru model of the first chapter consists of about 30 plans.

2.2. Asbru in KIV

KIV [4] is a tool for formal systems development. It provides strong proof support for all validation and verification tasks. KIV can handle large scale formal models by efficient proof techniques, multi-user support, and an ergonomic user interface. The specification language of KIV is based on higher-order algebraic specifications. Support to verify concurrent systems in temporal logic [3] [2] is included. Special feature of this temporal logic support is the explicit consideration of the environment. A *system description* is used to describe the state changes allowed by the TL system, while an *environment assumption* restricts the environmental behaviour. System and environment transitions alternate.

We use this formalisation to model the Asbru plan hierarchy as the system. All human interaction, like administration of drugs, measurement of relevant data and the like are modelled by the environment assumption.

In order to verify medical guidelines in KIV, an algebraic specification has been constructed for the Asbru language. This specification formally defines the semantics of Asbru. Based on this specification, an Asbru guideline can be directly translated to KIV. Figure 3 gives an example Asbru plan in KIV. The equation defines plan breast-conserving-therapy (bct). The plan is selectable, if patient-prefers-bct evaluates to true for a given



(a) Asbru semantics

```
considered#(sk, sk0; var State)
begin
  if plan-not-filter(sk, State) then
    State[sk] := rejected
    else if plan-filter(sk, State) then
      State[sk] := possible;
      possible#(sk, sk0; State)
    else
      State[sk] := considered;
      considered#(sk, sk0; State)
  end
```

(b) KIV implementation

Figure 4. Dynamic behaviour of Asbru

patient data pd (filter condition). Other conditions are defined as default conditions. The plan body contains three sub-plans (bct1, mcte, dcis), which are executed in sequential order. The sub plans are optional. This can be seen from the expression wait-for-n(0, bct1 + mcte + dcis + []), which states, that none of the sub-plans bct1, mcte and dcis have to be completed for the plan bct to complete. We have taken special care to define Asbru plans in KIV such that translation from Asbru to KIV is almost one-to-one. As a result, the translation process from Asbru to KIV is fully automatic. The algebraic specification for Asbru defines the formal semantics of functions mk-asbru-def, mk-aasbruc, etc. More details can be found in [13].

An example for the implementation of the dynamic behaviour of Asbru plans can be found in Figures 4(b) and 4(a). Here we present excerpt from the state-chart in Figure 1. The close match of the considered# procedure as it is implemented in KIV (seen in Figure 4(b)) and the semantics definition of Asbru in Figure 4(a) is obvious.

2.3. Properties

Properties can be expressed in full first order (linear) temporal logic. For the formulation of properties a special translation scheme has been devised [18], such that the informal properties are formalised step-by-step in a transparent way, enabling medical experts to supervise the translation and spot modelling anomalies at the earliest possible moment.

Proof support for these properties has been added based on symbolic execution with induction: an Asbru guideline is executed step by step and induction is applied if execution loops. An example proof obligation is shown in Figure 5. Our goal is to verify this property, which states, that as long as mcte remains activated the parameter more-than-focal-tumour-involvement is false.

Different variables are used for the formulation of properties as well as the implementation of the semantics of Asbru. For example a variable designated AS maps asbru plans to plan states. It is used by selecting the plan state via the plan name, so the term AS[mcte] represents the current planstate of plan mcte. The variable PDH summarises all medical knowledge, like the blood pressure, age or tumour stage of a patient. It is accessed by a time, which makes it possible not only to base decisions on the current medical state but also on the patient medical history.

$$
\begin{array}{ll}
 & \textit{/* system configuration */} \\
 & AS[\text{mcte}] = \text{inactive}, \\
 & \textit{/* system behaviour */} \\
 & [\text{inactive\#}(\text{mcte}, sk;\ \mathit{Tick}, PDH, AC, AS, PC, \ldots)], \\
 & \textit{/* environment assumption */} \\
 & \Box\ (AS''[\text{mcte}] = AS'[\text{mcte}] \wedge AS''[\text{sp}] = AS'[\text{sp}] \wedge \ \ldots) \\
\vdash & \textit{/* property */} \\
 & \Box\ (\quad AS'[\text{mcte}] = \text{activated} \wedge \ \ldots \\
 & \qquad \to \neg\ PDH[AC].\text{more-than-focal-tumour-involvement})
\end{array}
$$

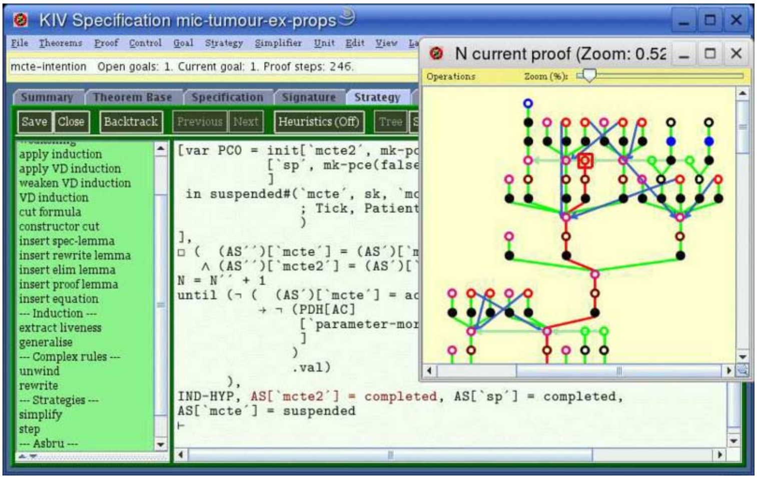
Figure 5. Example proof obligation in KIV

For the verification a temporal logic is used, in which system transitions and environment transitions alternate. Variables therefore maybe unprimed to describe their initial states, single primed to describe the state after the system transition and before the environment transition. Finally variables may be doubly primed, designating the state after the environment transition. Therefore a formula like AS''[mcte] = AS'[mcte] states that the environment transition does not change the planstate of the plan mcte.

The proof obligation consists of four parts. The first part is the system configuration, which expresses the relevant part of the current state. In the example this is the status of the plan mcte, which is in the inactive state (compare to Figure 1). Also, the consider signal has been sent, allowing the plan to proceed to the considered state. Next in the property is the system behaviour. This is a description of all the procedures which are currently enabled and possibly changing the system state. In this initial state, only the procedure representing the inactive mcte plan is present. As already mentioned, system and environment transitions alternate. In this example, it is necessary to restrict the environment transition such that it is not allowed to change the state of the plans mcte and sp. Further restrictions are required but omitted in Fig. 5 for better readability. Last part of the proof obligation is the property. In the example it states, that as long as the plan mcte is activated, the parameter more-than-focal-tumour-involvement is false.

3. Example

Verification of the example property starts with the Asbru plan mcte being inactive and the consider signal being sent. The procedure inactive#$(\ldots)$ defines the system behaviour for an inactive Asbru plan. This procedure implements part of the plan state model of Figure 1. This procedure is a model of the system behaviour as is sketched in Section 2.2. Technically the procedure relates unprimed and single primed temporal variables. The environment, relating single primed and double primed variables is restricted by the environment assumption. We assume here, that internal communication of the Asbru plans, i.e. signals sent by mcte to its sub-plans and from them further down the hierarchy, is left untouched by the environment. Furthermore, the Asbru state of all the plans must not be changed by the environment. This assumption is required because an Asbru plan bases decisions about its termination on the states of the sub-plans. If the environment was allowed to change these states at will, mcte might consider one of its sub-plans to be in a different state, than it actually is.



mcte-intention Open goals: 1. Current goal: 1. Proof steps: 246.

```
[var PCO = init[`mcte2´, mk-pc
             [`sp´, mk-pce(false
             ]
 in suspended#(`mcte´, sk, `mc
               ; Tick, Patient
               )
],
□ (  (AS´´)[`mcte´] = (AS´)[`m
   ∧ (AS´´)[`mcte2´] = (AS´)[`
N = N´´ + 1
until (¬ (  (AS´)[`mcte´] = ac
          → ¬ (PDH[AC]
               [`parameter-mor
               ]
              )
              .val)
        ),
IND-HYP, AS[`mcte2´] = completed, AS[`sp´] = completed,
AS[`mcte´] = suspended
⊢
```

Figure 6. Example proof in KIV

In order to prove the property, the Asbru plan is executed step by step. In the first state, plan mcte is inactive. Execution adheres to the plan state model of Figure 1. After the first transition, plan mcte is considered. If the filter condition F is satisfied, then the plan is possible in the next state, if it is no longer satisfiable, the plan is rejected. Evaluation goes on and once the plan is activated, its sub plans start their evaluation, changing their states to considered and so on. Execution results in a proof tree depicted on the right hand side of Figure 6. The final tree represents all of the possible paths of execution. For infinite execution paths, induction is used such that the overall proof tree is always finite. During execution, the current state is displayed in the large area of the main proof window. In Fig. 6, plan mcte is currently suspended while execution of the sub plans have been completed. On the left hand side of the main proof window, a list of applicable proof rules are displayed, the most important rules being *step* to execute a system step and induction rules to reason with loops. System execution is automatic to a large extent, however, invariants for induction must be manually provided.

In contrast to interpretation of Asbru guidelines, the proof engine accepts symbolic values for the variables. For example, the patient state can be described with arbitrary first order formulas. Thus, proofs can be constructed for arbitrary patient input. KIV offers strong support for reasoning in first order logic. Details on verifying Asbru in KIV can be found in [13].

4. Outlook

The proof method presented here is only one of four ingredients to interactively validate a medical guideline. In this section, we shortly present other aspects of medical guideline validation.

4.1. Proof Decomposition

Experiments have shown, that the proof technique described here can be used nicely up to approximately 10 concurrently running Asbru plans, which is a severe limitation, if faced with a case study with a total of more than 30 plans. For those rather large hierarchies, proofs have to be decomposed.

Proving a property, part of the hierarchy required to prove this property can be abstracted by the relevant behaviour. For example, in a proof it may only be important, that a certain plan always completes and never gets aborted or rejected. If then, for example, the ordering of sub-plans or the set of started sub-plans are irrelevant for the correctness of the original property, a separate property can be formulated, claiming that the said plan always completes. This property then abstracts from the exact execution path, the amount and order of started children. This approach has been described in detail in [13].

4.2. Background Knowledge

Asbru plans and Asbru verification is independent of certain medical knowledge to a very far degree. Only little about the interaction between different drugs or even the effects of single drugs on humans is modelled in Asbru. The result of this is, that properties which are verifiable are mainly focussed on the treatment process, not the outcome of a treatment.

Incorporation of background knowledge changes this. The idea is to describe a more fine grained model of the patients behaviour, especially the relation of behaviour and the intake of drugs. With this, the range of verifiable properties widens. It is not only possible to establish that certain process provisions are adhered to by the guideline. In addition to this, it is possible to check, whether the treatments are more invasive than necessary, for example, whether the guideline forces surgery, while a drug therapy would have sufficed. Research on this topic has been described in [7].

4.3. Counter Example

Often verification of medical properties does not succeed. There are cases in which the verification will result in a non-satisfiable proof obligation.

In these cases, it is interesting to examine, what exactly has happened. Usually, it is possible to determine the circumstances which led to the violation of the property. Also, the violation can be traced back with some effort, finally pointing to a plan control, a condition or a wait-for statement, which is the suspected cause for the violation. To harden this assumption, a separate proof is done, in which it is verified to be correct, i.e. the suspected side conditions indeed lead to a violated property. This is currently only possible with a subset of counter examples and is a research question being worked on.

5. Summary

In our approach it is possible to formulate a wide variety of properties. However, certainty of medical knowledge reduces the number of properties that actually can be verified. Typical properties, which can be tackled without difficulties include progress and ordering properties. Examples are statements like ‘this drug is administered at most twice’ or ‘treatment a is only administered after treatment b succeeded’.

Harder are properties relying on medical background knowledge. Properties like the invasiveness of treatments have been verified in a case study concerned with the treatment of diabetes mellitus type II. There, excessive medical background knowledge about the consequences of the administration of different drugs has been formalised and has been used to verify, that a guideline would always prescribe the least invasive treatment possible. Most times, however, the cause-consequence analysis of drug administration is much harder. In many cases, drugs do not have the same effects on different people and sometimes the effects of drugs are not predictable. As soon as probabilistic properties are involved, this approach reaches its limitations.

The informal guideline leaves room for interpretation, sometimes intentionally, for example because the scientific evidence is inconclusive. With the formal verification, we are able to uncover

We consider the formal verification therefore to have the following positive effects:

(I) Special cases are made explicit and visible

contains many implicit ambiguities. Formalising the guideline forces

6. Conclusion

In KIV, we have implemented support for the verification of medical guidelines modelled in Asbru. The proof support is based on an intuitive proof strategy: symbolic execution with induction. The proof method is automated to a large extent. The guidelines are executed step by step and induction is used to reason about loops. Large guidelines can be verified including complex data types and time annotations. This complements the use of model checking to easily verify simple properties of smaller guidelines with state-finite data types.

The implementation has been evaluated in several case studies: a guideline from the American Academy of Pediatrics for the treatment of jaundiced newborns, a Dutch general practitioners guidelines for diabetes mellitus type II, and a Dutch guideline for the treatment of breast cancer. With KIV, we have found a number of errors in the Asbru models, in the formal semantics of Asbru, and in the KIV implementation.

A detailed comparison of model checking and interactive verification of medical guidelines remains for future investigation.

References

[1] M. Balser, C. Duelli, and W. Reif. Formal semantics of Asbru – An Overview. In *Proceedings of IDPT 2002*. Society for Design and Process Science, 2002.

[2] M. Balser, C. Duelli, W. Reif, and G. Schellhorn. Verifying concurrent systems with symbolic execution. *Journal of Logic and Computation*, 12(4):549–560, 2002.

[3] Michael Balser. *Verifying Concurrent System with Symbolic Execution – Temporal Reasoning is Symbolic Execution with a Little Induction*. PhD thesis, University of Augsburg, Augsburg, Germany, 2005.

[4] Michael Balser, Wolfgang Reif, Gerhard Schellhorn, Kurt Stenzel, and Andreas Thums. Formal system development with KIV. In T. Maibaum, editor, *Fundamental Approaches to Software Engineering*. Springer LNCS 1783, 2000.

[5] S. Bäumler, M. Balser, A. Dunets, W. Reif, and J. Schmitt. Verification of medical guidelines by model checking: a case study. In *Proc. of the 13th International SPIN Workshop on Model Checking of Software*, number 3925 in LNCS, pages 219–233. Springer-Verlag, 2006.

[6] J. Fox, N. Johns, and A. Rahmanzadeh. Disseminating medical knowledge – the PROforma approach. *Artificial Intelligence in Medicine*, 14, 1998.

[7] A. Hommersom, P. Groot, P. Lucas, M. Balser, and J. Schmitt. Combining task execution and background knowledge for the verification of medical guidelines. In M. Bramer, F. Coenen, and A. Tuson, editors, *Proceedings of AI-2006, the 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, number XXIII in Research and Development in Intelligent Systems, pages 3–16. Springer, 2006.

[8] S. Miksch, Y. Shahar, and P. Johnson. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. In E. Motta, F. v. Harmelen, C. Pierret-Golbreich, I. Filby, and N. Wijngaards, editors, *7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*. Milton Keynes, UK, 1997.

[9] M. Musen, S. Tu, A. Das, and Y. Shahar. EON: A component-based approach to automation of protocol-directed therapy. *Am Med Inform Assoc*, 3, 1996.

[10] L. Ohno-Machado, J. Gennari, S. Murphy, N. Jain, S. Tu, D. Oliver, and E. Pattison-Gordon. The guideline interchange format: a model for representing guidelines. *Am Med Inform Assoc*, 5, 1998.

[11] I. Purves, B. Sugden, N. Booth, and M. Sowerby. The PRODIGY project–the iterative development of the release one model. In *Proc AMIA Symp*, 1999.

[12] S. Quaglini, M. Stefanelli, A. Cavallini, G. Micieli, C. Fassino, and C. Mossa. Guideline-based careflow systems. *Artifical Intelligence in Medicine*, 20, 2000.

[13] J. Schmitt, M. Balser, and W. Reif. Asbru in KIV v2.1 – a tutorial. Technical Report 2006-03, University of Augsburg, 2006.

[14] J. Schmitt, W. Reif, A. Seyfang, and S. Miksch. Temporal dimension of medical guidelines: The semantics of Asbru time annotations. In *Proceedings of Workshop on AI techniques in healthcare: evidence-based guidelines and protocols (ECAI'06)*, 2006.

[15] Radu Serban, Annette ten Teije, Frank van Harmelen, Mar Marcos, and Cristina Polo-Conde. Extraction and use of linguistic patterns for modelling medical guidelines. *AI in Medicine*, 39(2):137–149, 2007.

[16] Andreas Seyfang, Silvia Miksch, Mar Marcos, Jolanda Wittenberg, Cristina Polo-Conde, and Kitty Rosenbrand. Bridging the gap between informal and formal guideline representations. In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *ECAI*, pages 447–451. IOS Press, 2006.

[17] Andreas Seyfang, Silvia Miksch, Cristina Polo-Conde, Jolanda Wittenberg, Mar Marcos, and Kitty Rosenbrand. Mhb - a many-headed bridge between informal and formal guideline representations. In *AIME*, pages 146–150, 2005.

[18] Ruud Stegers. From natural language to formal proof goal: Structured goal formalisation applied to medical guidelines. Master's thesis, Vrije Universiteit, Amsterdam, 2006.

[19] Marjolein van Gendt, Annette ten Teije, Radu Serban, and Frank van Harmelen. Formalising medical quality indicators to improve guidelines. In *AIME*, pages 201–210, 2005.

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-263

Improving the Execution of Clinical Guidelines and Temporal Data Abstraction in High-Frequency Domains

Andreas SEYFANG [a,1], Michael PAESOLD [a], Peter VOTRUBA [a] and Silvia MIKSCH [a,b]

[a] *Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria*

[b] *Department of Information & Knowledge Engineering, Danube-University Krems, Austria*

Abstract. The execution of clinical guidelines and protocols (CGPs) is a challenging task in high-frequency domains such as Intensive Care Units. On the one hand, sophisticated temporal data abstraction is required to match the low-level information from monitoring devices and electronic patient records with the high-level concepts in the CGPs. On the other hand, the frequency of the data delivered by monitoring devices mandates a highly efficient implementation of the reasoning engine which handles both data abstraction and execution of the guideline.

The language Asbru represents CGPs as a hierarchy of skeletal plans and integrates intelligent temporal data abstraction with plan execution to bridge the gap between measurements and concepts in CGPs.

We present our Asbru interpreter, which compiles abstraction rules and plans into a network of abstraction modules by the system. This network performs the content of the plans triggered by the arriving patient data. Our approach evaluated to be efficient enough to handle high-frequency data while coping with complex guidelines and temporal data abstraction.

Keywords. Guideline Execution, Decision Support Systems

Introduction

In the field of medicine, the application of clinical guidelines and protocols (CGPs) helps to improve the quality of care by ensuring the optimal choice of treatment. CGPs are "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" [4]. Such guidelines are based on the best empirical evidence available at the moment. A guideline describes the optimal care for patients and therefore, when properly applied, it is assumed that they improve the quality of care.

[1] Corresponding Author: Andreas Seyfang, Institute of Software Technology and Interactive Systems, Vienna University of Technology, Favoritenstraße 9-11/E188, A-1040 Vienna, Austria; E-mail: seyfang@ifs.tuwien.ac.at.

A precondition for the successful application of CGPs is automatic abstraction of context-dependent time-annotated raw-data (e.g., percent of oxygen in blood at a certain second) to high-level medical concepts (e.g., sufficient oxygen saturation during an extended period of observation). This is performed by temporal data abstraction. Therefore, the integration of temporal data abstraction with the execution of CGPs is a precondition for the successful application of computer support for CGP execution.

In the following, we discuss related work in Section 1. The language used to represent a CGP in computer-executable form is Asbru. It is described in Section 2. In Section 3 we explore the details of the interpreter, illustrated by an example. The evaluation and future work is described in Section 4 followed by the conclusion.

1. Related Work

In the past, different CGP frameworks were introduced (compare [15] for a detailed comparison and [11] for a pattern-based analysis of CGPs). Asbru [17] is one representative and scored favourably in both comparisons. CGP languages not covered there are GLARE [18], containing ideas similar to Asbru, and CIGDec [12] taking a declarative approach which can be emulated easily by Asbru constructs (unordered and cyclical plan and temporal constraints in the plan activation as desired).

There have already been two attempts to implement an execution engine for Asbru. The first implementation created by Bosse [3] translated the guideline into a representation suitable to be executed in Clips. The implementation was customised for a single CGP and was therefore abandoned after the end of the project. A more general implementation of Asbru is AsbruRTM [6]. It has been used to test Asbru guidelines in Intensive Care Units [7]. Unfortunately, AsbruRTM only supports a subset of the available plan types in Asbru and does not integrate advanced (temporal) data abstraction.

Spock [19] is a system for application of guidelines in Hybrid-Asbru, which is a semi-formal guideline language that combines formal structure with description text. Spock is therefore not suited for fully-automated execution, but it can support a human agent applying a guideline. Spock is integrated with the IDAN architecture [2] and can utilise its temporal abstraction capabilities for decision support.

None of the above systems closely integrate plan execution and the required data abstraction into the clinical data flow in a high frequency domain such as intensive care. We therefore designed a seamless framework for the abstraction of data, the execution of plans and monitoring the environment for relevant changes [14,16] as described in Section 3. It is complemented by a range of tools for the authoring and visualisation of various aspects of the guideline and the abstraction rules [1].

2. Representing Plans and Abstractions in Asbru

Asbru is a time-oriented plan representation language that describes CGPs as skeletal plans [17]. Skeletal plans are plan schemata at various levels of detail, capturing the essence of the procedure, but leave room for execution-time flexibility in the achievement of particular goals [5]. Several knowledge roles are attached to a plan: preferences, intentions, conditions, effects, and a plan body, which describes the actions to be taken.

Asbru's distinguishing features are that (1) intentions, conditions, effects, and world states are temporal patterns, which allow reasoning about the contained knowledge; (2) actions and states can be continuous (durative); (3) the language allows to model temporal uncertainties, different granularities, and repeated patterns in events, actions, plans, and world states; (4) plans are executed in sequence, all plans or some plans in parallel, or unordered with or without mutual exclusion; (5) because of the advanced (temporal) data abstraction capabilities, diagnosis and treatment can be tightly integrated allowing each one to support the other one.

All conditions for the transition from one plan state to another are expressed in terms of temporal patterns. A temporal pattern consists of one or more parameter propositions or plan-state descriptions. Each parameter proposition contains a value description, a context, and a time annotation. The time annotation used allows a representation of uncertainty in starting time, ending time, and duration of an interval. Start and end are defined as shifts from a reference point. Reference points can be defined as sets of cyclical time points or references to parameter changes, allowing repeated temporal patterns.

Asbru differentiates between seven plan states: *considered*, *possible* and *rejected* represent the plan-selection phase, *activated*, *suspended*, *completed*, and *aborted* represent the plan execution phase.

Asbru plan libraries are written in XML and consist of two major parts, the domain definition section and the plans section. The plan section contains the plan definitions as described above. The domain definition defines both quantitative and qualitative parameters. These can be directly input or abstracted from other parameters. There is a wide range of abstractions available. They can be grouped into value abstractions which exclusively deal with the value dimension of measurements, and value aggregations where temporal abstractions are applied on the measurement. The result of the abstraction is accessed in the plans – both in conditions and assignments.

3. The Asbru Interpreter

Conceptually, the Asbru Interpreter consists of three basic units: data abstraction, monitoring, and plan execution. In the data abstraction unit, various temporal or atemporal abstractions are applied to the patient data to gain information at higher conceptual levels. The provided quantitative or qualitative data is monitored to detect temporal patterns in the abstracted data. This information is used to control the selection and execution of plans. This data flow is not unidirectional, instead, the execution unit can interact with both monitoring and abstraction unit to adjust the monitored patterns and to adapt the abstraction process to the context given by the current plan states.

The interpreter has two different modes of operation: Batch mode and online mode. In batch mode, a large set of records is read to validate a guideline against patient data or to create complex abstractions of the data for later analysis. In online mode, data can be read from monitoring devices in addition to the user input.

3.1. System Architecture

Figure 1 shows the parts of the interpreter on the implementation level. The main parts are the Asbru Compiler and the Execution Manager. The three conceptual components

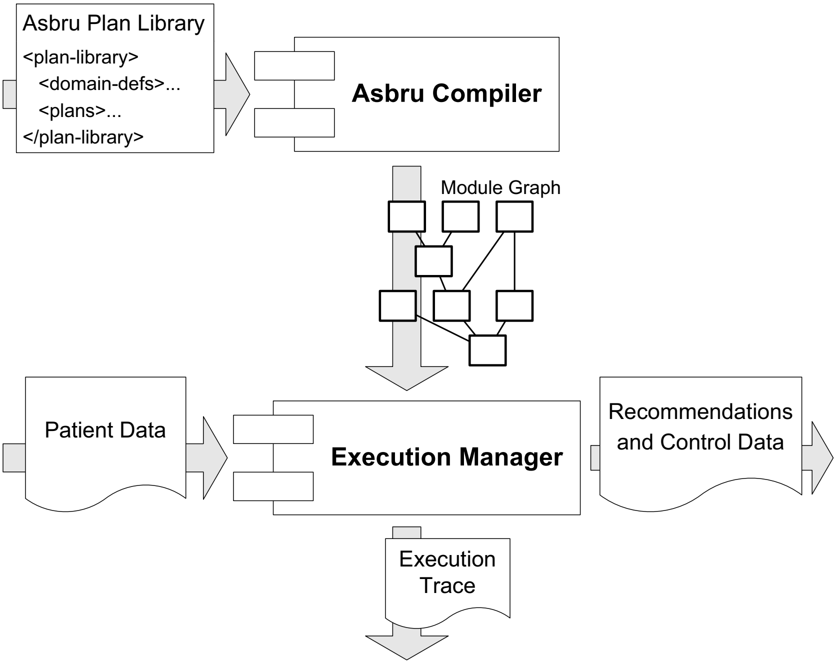


Figure 1. System architecture and data flow in the Asbru Interpreter. The Asbru plan library is compiled into a directed graph of modules by the Asbru Compiler. The Execution Manager uses this module network to process patient data and execute the plans representing the guideline. The output of the modules is provided in a uniform way, to be displayed in a graphical user interface. In addition, all state transitions are documented in a log file which is then translated to various table formats to ease the analysis of the execution path.

mentioned above – data abstraction, monitoring, and plan execution – are seamlessly integrated in the module graph.

At program start-up, the Asbru plan library XML file is compiled into a directed graph of modules. For each time step, the Execution Manager enacts each of the modules in the network to process patient data, monitor temporal patterns, and execute the plans in the guideline. These modules are largely compatible with each other, which allows information extracted by any module to flow back into the abstraction or monitoring process. To handle complex networks with many inputs in high-frequency applications, the Execution Manager ensures that each module is enacted exactly when needed, allowing for small time steps by some modules without the overhead created by other modules which would not provide new information at that moment.

The output of the system can be displayed in a graphical user interface. In addition an interactive, graphical user interface currently under development will allow the stepwise control of plan execution. Using custom-built modules, the output can be integrated with the control of medical devices in a close-loop setting.

During operation, the interpreter writes an extensive log file documenting all abstraction steps and plan state changes. This is later transformed into various reports focussing on different aspects by easily customisable post-processing tools. This is of particular importance when validating a CGP against a set of patient data, which is the current main application of the interpreter.

3.2. Module Design

Each Asbru statement is translated to one or more modules. Some of these modules are simple, while others implement complex logic. At each time step, a module receives zero, one, or more data points as input from its precursors in the abstraction graph and generates zero or one output data point. This output can be a simple time-stamped numeric value, or a complex structure such as the linear regression of a series of measurements, or the state of a plan together with synchronisation tokens for its children and parent plan. An important group of the modules – the temporal aggregation modules – produces output at a lower rate than its input, thus relieving its successors in the abstraction graph from the larger part of the data load by outputting high-level concepts such as “sufficient oxygen” only.

Modules can set alarms, to be triggered when a certain span of time is elapsed. Here we distinguish between *pre-alarms* and *post-alarms* depending on whether the alarm is triggered before or after processing the data for this time step. This distinction allows the implementation of both complex and convex temporal intervals. Alarms are set by various monitoring and value aggregation modules.

The available modules can be divided into the following categories.

Constant modules and system variables. These are the simplest category. They do not receive input. Constants modules are used to implement the constants found in an Asbru plan. System variables continuously produce values such as the current date.

Raw data modules. These modules interface the input channels and map the raw-data parameter definitions in Asbru. In online operation, they passively wait for arriving input data and return the next available value. In batch mode they read data from files.

Value abstraction modules. This group comprises the logical and arithmetic combination of inputs, and the mapping of quantitative values to qualitative categories.

Value aggregation modules. In order to map high-frequency, error-prone inputs to high-level concepts, it is mandatory to aggregate series of measurements and to derive the abstractions from them, and not from single measurements. Such aggregates can be descriptive statistics applied to moving time-windows, or more complex algorithms such as the *spread* [10].

Monitoring modules. These modules handle temporal patterns, such as parameter propositions, which control the state changes of Asbru plans. A parameter proposition has several states. Initially it is *not fulfilled.* As soon as an interval matches the time annotation, the parameter proposition becomes *fulfilled.* If the reference point is the symbolic value *now*, i. e., the current time point of evaluation, then the parameter proposition can become *no more fulfilled* in the future (provided that the condition or the context evaluate to true no more). A detailed discussion can be found in [14,16]. Other Asbru features which are also covered by modules of this category are temporal constraints or constraint combinations.

Temporal abstraction modules. The patterns detected by monitoring modules and aggregates of the measurements often need one or more steps of temporal abstraction to detect complex patterns in the input data such as “five episodes of apnoea followed by hyperoxemia during the previous hour”.

Plan modules. Modules in this category represent Asbru plans or single plan steps. The network of parent and child plans is fully integrated with the other modules to form the module graph. Thus, the output from plan modules can be fed back into further abstraction steps.

3.3. Module Interaction

The following actions are performed for each module: providing optional control data, attend to pre-alarms, process input data, attend to post-alarms, and finally store the output of this module for use by other modules down the abstraction stream.

A plan module that represents a logical decomposition of plans needs to synchronise the plan modules representing its sub-plans. It does this by sending special data points to these child plan modules. Since the communication of plans with their children introduces cycles in the module graph, it is necessary to invoke plan modules more than once per data point received from the environment. Therefore, we divide each external *macro time step* into several internal *micro time steps*. This means that in-between processing two succeeding data points, the internal state propagation is propagated first. This does not introduce additional overhead as only the few concerned modules consume compute time in this phase.

The execution manager ensures that each module is only enacted upon new input. Therefore, a large number of complex low-frequency abstractions can be closely coupled with high-frequency modules. Further details are given in [14].

The input and output of all modules in the abstraction graph is a subclass of time-stamped data point. In simple cases, the value of the data point is a number or a qualitative value. In other cases, it is an abstract concept such as a linear regression computed for a sliding time window. Plan states are also communicated as time-stamped data points, as well as trigger messages between parent and child plans which control the order in which plans are executed. This means that both the (temporal) abstraction of data and the execution of plan steps use the same framework.

There are two major benefits from this approach. First, the output from plans can be feed into further temporal abstraction modules which again may provide input to other plans. Second, observations regarding plans, e.g., their duration or state, are available to abstraction modules, which can combine them with input data or other abstractions, implementing the seamless integration of reasoning about patient state and reasoning about plan state (i.e., treatment steps).

3.4. Example: Ventilation in Neonates

From the field of artificial ventilation in neonates, we extract the following fragment of a protocol controlling the fraction of inspired oxygen based on measurements of partial pressure on oxygen in blood.

> *"An external monitoring device measures the saturation of oxygen in blood SpO2. It delivers numeric values at a rate of 1 Hz. These values are abstracted to qualitative values. E.g., SpO2 below 80% is mapped to* apnoea*, while higher values are mapped to* decreased*,* normal *and* increased*. If the qualitative value of SpO2 equals* apnoea *for at least 4 seconds, then normal ventilation should be suspended. In this situation, the patient will receive emergency treatment by the medical staff. If the patient returns to less critical state, as defined by SpO2 being unequal* apnoea *for at least 10 seconds, normal ventilation is resumed."*

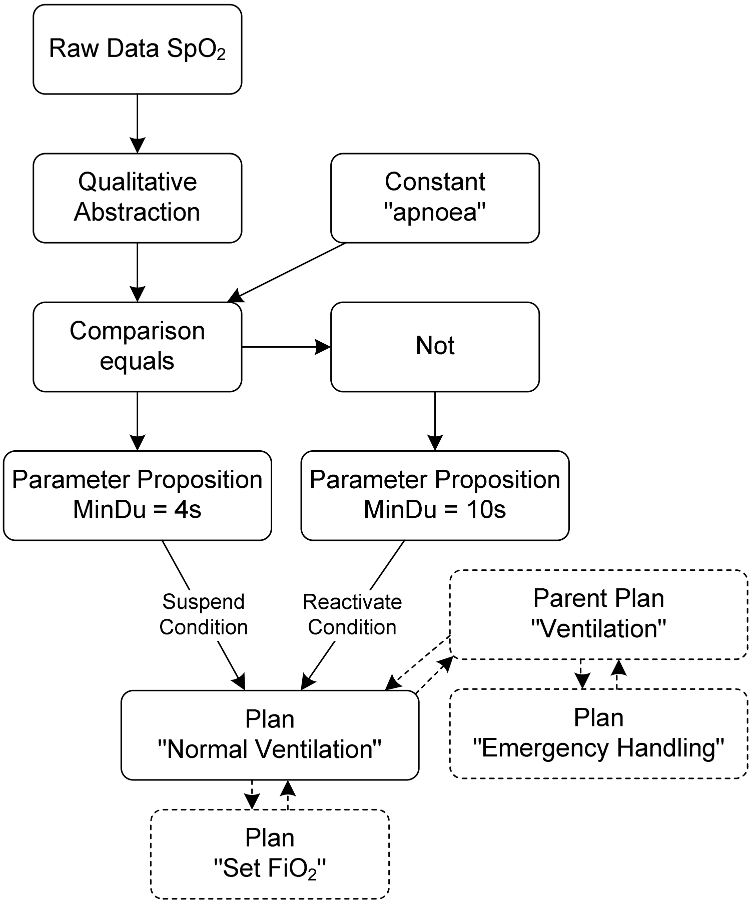

Raw Data SpO_2
Qualitative Abstraction
Constant "apnoea"
Comparison equals
Not
Parameter Proposition MinDu = 4s
Parameter Proposition MinDu = 10s
Suspend Condition
Reactivate Condition
Parent Plan "Ventilation"
Plan "Normal Ventilation"
Plan "Emergency Handling"
Plan "Set FiO_2"

Figure 2. Sample module graph. The module graph maps directly to the described Asbru model. A raw data module has been created for the parameter `SpO2`, which is connected to a qualitative abstraction module. The resulting qualitative value is compared against a qualitative constant. This comparison module is connected parameter proposition module used for the suspend condition of the *normal ventilation* plan module. Another parameter proposition module is connected to the re-activate condition, which uses the negated output of the comparison module as input (because the re-activate condition demands that the qualitative abstraction of SpO2 be not equal to 'apnoea'). The three other plan modules framed with a dashed line are shown to illustrate the context of this example, but will not be further explained in this paper.

In Asbru, `SpO2` is a raw parameter. In addition, we introduce an abstracted qualitative parameter *SpO2-qualitative* with the possible values *apnoea*, *decreased*, *normal* and *increased*, where `apnoea` corresponds to *SpO2* $<$ *80%*. Furthermore, we create a plan called *normal-ventilation* with a suspend condition and a re-activate condition, both realised using parameter propositions. Each parameter proposition has reference point *now*. The first parameter proposition has the condition *SpO2-qualitative equal apnoea* and the time annotation *minimum-duration* = *4 sec*. The second parameter proposition contains *SpO2-qualitative not-equal apnoea* and *minimum-duration* = *10 sec*.

Figure 2 shows the relevant part of the module graph that is generated by the Asbru Compiler based on this specification. In terms of the overall concept, the monitoring task is performed by the parameter proposition modules, the data abstraction is provided

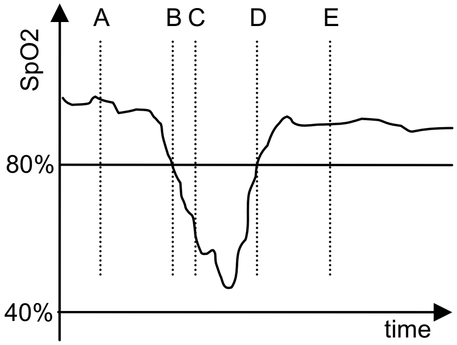


Figure 3. Sample input data. The graph shows the measurements of saturation of oxygen in blood (SpO2). The horizontal line shows the threshold between the qualitative region of apnoea (below 80%) and the other qualitative regions not relevant in our example. The dotted lines mark relevant time points described in the text.

by the modules feeding them, and the modules shown below the parameter proposition modules implement the plan execution.

In the following we discuss a typical series of events during abstracting and monitoring the input using the described modules. Figure 3 shows an excerpt from monitored saturation of oxygen in blood (SpO2).

Time point A represents one of many time steps during which the plan *normal ventilation* is activated and no changes are required. The value of `SpO2` is above the threshold (80%). Therefore, the comparison module does not create new output, after outputting *false* once at program start.

At time point B, the SpO2 value falls below the threshold. Therefore, the comparison module outputs `true`. The directly connected parameter-proposition module at the left detects a positive flank, i. e., a change from `false` to `true` in its input channel and sets an alarm to *current-time + 4s*.

At time point C (i. e., 4 seconds after B), the alarm set at time point B triggers and since the value of SpO2 did not change (i. e., no negative flank occurred), the parameter proposition module reports a found episode to the plan module. This means that the suspend condition of the plan gets fulfilled and the plan changes its state to `suspended`.

At time point D, the first parameter-proposition detects a negative flank (SpO2 is no more in the range of apnoea, therefore the value of the input changed from `true` to `false`) and changes its output to `no-more-fulfilled`. In this case, this input has no consequence for the state of the plan module. The second parameter proposition module (which is monitoring "qualitative SpO2 not equal apnoea) detects a positive flank, as the comparison module outputs*false* now which is inverted by the `not` module. Consequently, this parameter proposition module sets an alarm to *current-time + 10s*.

At time point E (i. e., 10 seconds after D), the previously set alarm for the second parameter proposition module triggers and since there was no negative flank for this module until then, this parameter-proposition reports now a found episode to the plan module. Therefore, the re-activate condition of the plan evaluates to `true` and the plan resumes (i. e., it changes its state back to `activate`).

If a negative flank would appear between D and E, i.e., if another apnoea would start, then the alarm for the second parameter proposition would be cancelled. Later, on another positive flank, the alarm will be set anew (to 10 seconds after that flank).

4. Evaluation and Future Work

A real-world guideline for breast cancer [13] was modelled in Asbru and test runs of the resulting model in the interpreter were successful. Running the interpreter on patient cases will allow the comparison between the expected outcome of applying the guideline and the actual outcome according to the model in Asbru.

This form of *validation* of the guideline is a very important link between the *verification* of the model, which takes today's verifiers to their limits because of the complexity of the model, and the discussion of prototypical execution paths with domain experts, which can only cover a small fraction of all possible paths.

Besides dealing with the low-frequency domain of breast cancer, practical tests with high-frequency data showed that the interpreter processes input from 10 channels and moderately complex abstractions thereof at a rate of more than 1 kHz on a standard PC. Most clinical data is recorded at 1 Hz, or 200 Hz. We therefore conclude that the computational performance is sufficient for online applications in clinical monitoring. Still, our implementation cannot be considered real-time software in the narrow sense – it is not possible to guarantee a certain response time under all circumstances.

Future work will go into the construction of dedicated modules to interface equipment at intensive care units. Similar modules will allow the integration with other temporal abstraction systems. In addition, a graphical user interface to allow the interactive use of the interpreter by non-computer experts is under development.

5. Conclusion

Plan execution in real-world high-frequency domains such as intensive care units demand for tight integration of temporal data abstraction and plan execution to achieve the required intelligent reaction to unpredictable changes in the environment, i.e., the patient state.

While the knowledge in such domains is abstract, partly vague, or incomplete, and often complex, the data arrives at a high rate and in a format that is far from the level in which the domain knowledge is specified. Translating the domain knowledge to such low levels by a knowledge engineer leads to well known short-comings regarding maintenance and assuring the correctness of the model.

We therefore designed an interpreter, which takes a high-level specification of skeletal plans and temporal data abstraction and compiles them into a network of abstraction modules. Using elaborate management of data flow, these modules process the data at a high rate, even in complex configurations.

Tests have demonstrated that the interpreter can handle a complete guideline, large sets of patient records, and high-frequency measurements. Applications to large sets of data are currently in progress.

Acknowledgement

We wish to thank the members of the Protocure II project, who provided us with essential feedback about using the Asbru Interpreter.

This work is part of the Protocure II project, which is supported by the European Commissions IST program, under contract number IST FP6-508794.

References

[1] W. Aigner and S. Miksch, ‘Supporting protocol-based care in medicine via multiple coordinated views’, in *Proceedings of the Second International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2004)*, pp. 118–129. IEEE Computer Society Press, (2004).

[2] D. Boaz and Y. Shahar, ‘A framework for distributed mediation of temporal-abstraction queries to clinical databases’, *Artif Intell Med*, **34**(1), 3–24, (2005).

[3] T. Bosse, *An Interpreter for Clinical Guidelines in Asbru*, Master’s thesis, Vrije Universiteit Amsterdam, 2001.

[4] M. J. Field and K. H. Lohr, eds. *Clinical Practice Guidelines: Directions for a New Program*, Institute of Medicine, Washington DC, 1990. National Academy Press.

[5] P. E. Friedland and Y. Iwasaki, ‘The concept and implementation of skeletal plans’, *Journal of Automated Reasoning*, **1**(2), 161–208, (1985).

[6] C. Fuchsberger, *Entwicklung einer Ausführungseinheit für Asbru Light*, Master’s thesis, Vienna University of Technology, 2003.

[7] C. Fuchsberger, J. Hunter, and P. McCue, ‘Testing Asbru guidelines and protocols for neonatal intensive care’, in *AIME 2005*, pp. 101–110. Springer, (2005).

[8] M. K. Goldstein, R. W. Coleman, S. W. Tu, R. D. Shankar, M. J. O’Connor, M. A. Musen, et al., ‘Operationalizing clinical practice guidelines amidst changing evidence: Athena, an easily modifiable decision-support system for management of hypertension in primary care’, *JAMIA*, **9**(6 Suppl 1), 11–16, (2002).

[9] W. Horn, ‘AI in medicine on its way from knowledge-intensive to data-intensive systems’, *Artif Intell Med*, **23**(1), 5–12, (2001).

[10] S. Miksch, A. Seyfang, W. Horn, and C. Popow, ‘Abstracting steady qualitative descriptions over time from noisy, high-frequency data’, in *Artificial Intelligence in Medicine*, ed., W. Horn et al., pp. 281–290, Berlin, (1999). Springer.

[11] N. Mulyar, W.M.P. van der Aalst, M. Peleg, ‘A Pattern-based Analysis of Clinical Computer-Interpretable Guideline Modelling Languages’, *Journal of the American Medical Informatics Association*, **14**(6), (to appear).

[12] N. Mulyar, M. Pesic, W.M.P. van der Aalst, M. Peleg, ‘Towards the Flexibility in Clinical Guideline Modelling Languages’, *BPM Center Report BPM-07-04*, BPMcenter.org, 2007.

[13] Nationaal Borstkanker Overleg Nederland (NABON), *Guideline for the treatment of breast cancinoma*, Van Zuiden Communications B.V., 2002.

[14] M. Paesold, *Monitoring Temporal Patterns in Guideline Based Care*, Master’s thesis, Vienna University of Technology, 2006.

[15] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. Greenes, R. Hall, P Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. Shortliffe, and M. Stefanelli, ‘Comparing computer-interpretable guideline models: A case-study approach’, *JAMIA*, **10**(1), (2003).

[16] A. Seyfang, *An Integrated System for Temporal Data Abstraction to Facilitate Guideline Execution and Knowledge-Based Data Analysis*, PhD dissertation, Vienna University of Technology, 2008. To appear.

[17] A. Seyfang, R. Kosara, and S. Miksch, ‘Asbru 7.3 reference manual’, Technical report, Vienna University of Technology, (2002).

[18] P. Terenziani, G. Molino, and M. Torchio, ‘A modular approach for representing and executing clinical guidelines’, *Artif Intell Med*, **23**(3), 249–276, (2001).

[19] O. Young and Y. Shahar, ‘The Spock system: Developing a runtime application engine for hybrid-Asbru guidelines’, in *AIME 2005*, pp. 166–170. Springer, (2005).

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-873-1-273

Applying Artificial Intelligence to Clinical Guidelines: the GLARE Approach

Paolo TERENZIANI[a1], Stefania MONTANI[a], Alessio BOTTRIGHI[a], Gianpaolo MOLINO[b] and Mauro TORCHIO[b]

[a] *DI, Univ. Piemonte Orientale, Via Bellini 25, Alessandria, Italy*
[b] *Lab. Informatica Clinica, Az. Ospedaliera S. G. Battista, C.so Bramante 88, Torino, Italy*

Abstract. We present GLARE, a domain-independent system for acquiring, representing and executing clinical guidelines (GL). GLARE is characterized by the adoption of Artificial Intelligence (AI) techniques in the definition and implementation of the system. First of all, a high-level and user-friendly knowledge representation language has been designed. Second, a user-friendly acquisition tool, which provides expert physicians with various forms of help, has been implemented. Third, a tool for executing GL on a specific patient has been made available. At all the levels above, advanced AI techniques have been exploited, in order to enhance flexibility and user-friendliness and to provide decision support. Specifically, this chapter focuses on the methods we have developed in order to cope with (i) automatic resource-based adaptation of GL, (ii) representation and reasoning about temporal constraints in GL, (iii) decision making support, and (iv) model-based verification. We stress that, although we have devised such techniques within the GLARE project, they are mostly system-independent, so that they might be applied to other guideline management systems.

Keywords. Representation, Acquisition, Consistent checking, Execution, Decision Support, Verification

Introduction

Clinical guidelines (GL) represent the current understanding of the best clinical practice, and are now one of the most central areas of research in Artificial Intelligence (AI) in medicine and in medical decision making. GL play different roles in the clinical process: for example, they can be used to support physicians in the treatment of diseases, or for critiquing, for evaluation, and for education purposes. Many different systems and projects have been developed in recent years in order to realise computer-assisted management of GL (see, e.g., the collections [1, 2, 3] and the comparisons in [4]; moreover, several approaches are described in this book).

Our contribution to this research area is represented by GLARE (Guideline Acquisition, Representation and Execution), a domain-independent prototypical system to acquire, represent and execute GL, which has been built starting from 1997 in a

[1] Corresponding Author: DI, Università del Piemonte Orientale "Amedeo Avogadro", Via Bellini 25/g, 15100 Alessandria, Italy; E-mail: terenz@unipmn.it.

cooperation with Azienda Ospedaliera San Giovanni Battista in Turin, one of the largest hospitals in Italy [5]. In this chapter, we overview some of the main features of GLARE, with specific focus on the adoption of advanced AI techniques. A more detailed treatment of the different topics can be found in the referenced publications about GLARE. After a sketchy description of the architecture (Section 1) and basic features (Section 2) of our system, we show how the adoption of AI techniques provides relevant advantages in GLARE, as regards resource-based contextualization (Section 3), treatment of temporal constraints (Section 4), decision-making support (Section 5) and model-based verification (Section 6). It is important to emphasize that, although we focus on GLARE, all of the methodologies discussed in Sections 3-6 are system independent, so that they could also be applied to other GL management systems in the literature.

1. Architecture of GLARE

The core of GLARE (see box on the left of Figure 1) is based on a modular architecture. CG_KRM (Clinical Guidelines Knowledge Representation Manager) is the main module of the system: it manages the internal representation of GL, and operates as a domain-independent and task-independent knowledge server for the other modules; moreover it permanently stores the acquired GL in a dedicated Clinical Guidelines Database (CG-DB). The Clinical Guidelines Acquisition Manager (CG_AM) provides expert-physicians with a user-friendly graphical interface to introduce the GL into the CG_KRM and describe them. It may interact with four databases: the Pharmacological DB, storing a structured list of drugs and their costs; the Resources DB, listing the resources that are available in a given hospital (it is therefore used to represent the context-dependent version of a GL); the ICD DB, containing an international coding system of diseases; the Clinical DB, providing a "standard" terminology to be used when building a new GL, and storing the descriptions and the set of possible values of clinical findings.

The execution module (CG-EM) executes a GL for a specific patient, considering the patient's data (retrieved from the Patient DB). The schema of the Patient DB mirrors the schema of the Clinical DB. Therefore, the inter-action with the Clinical DB during the acquisition phase makes it possible to automatically retrieve data from the Patient DB at execution time. CG-EM stores the execution status in another DB (CG Instances) and interacts with the user-physician via a graphical interface (CG-IM).

GLARE's architecture is open: new modules and functionalities can be easily added to the system if\when necessary. In the latest years, we have added new modules and\or methodologies in order to cope with automatic resource-based contextualization (ADAPT module), temporal reasoning (TR), decision making support (DECIDE_HELP), and model-based verification (VERIFY). While all the methodologies to cope with the above issues have been widely addressed (see sections 3-6), not all the prototypical software modules have been implemented yet. However, we envision a modular architecture, in which all such modules will be loosely coupled with the core of GLARE, as shown in the right part of Figure 1.

2. GLARE: Basic Features

In this section, we briefly summarize some of the basic features of GLARE (a detailed description is provided in [5]).

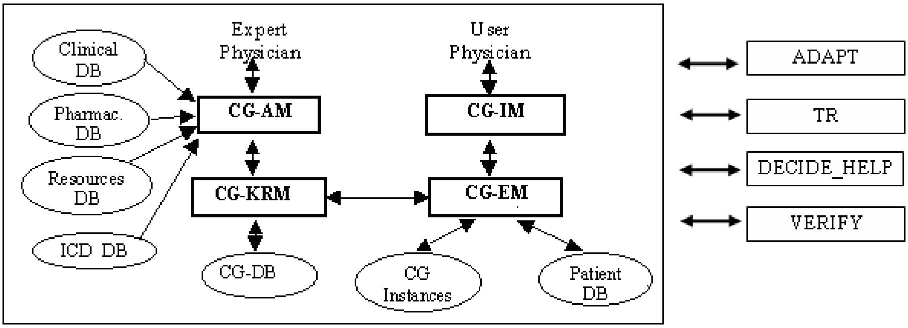


Figure 1. Architecture of GLARE. Rectangles represent computation modules, and ovals represents data/knowledge bases. The left box contains the "core" of GLARE, while additional modules in the right of the figure depict advanced AI support for several tasks.

Representation formalism. In the GLARE project, a GL is represented through the set of actions composing it. We distinguish between **atomic** and **composite actions**. Atomic actions can be regarded as elementary steps in a GL, in the sense that they do not need a further decomposition into sub-actions to be executed. Composite actions are composed by other actions (atomic or composite). Four different types of **atomic actions** can be distinguished in GLARE: *work actions, query actions, decisions* and *conclusions*. *Work actions* are atomic actions which must be executed at a given point of the GL, and can be described in terms of a set of attributes, such as name, (textual) description, cost, time, resources, goals. *Query actions* are requests of information, that can be obtained from the outside world (physicians, Databases, knowledge bases). *Decision actions* are specific types of actions embodying the criteria which can be used to select from alternative paths in a GL. In particular, *diagnostic decisions* are represented as an open set of triples <*diagnosis, parameter, score*> (where, in turn, a parameter is a triple <*data, attribute, value*>), plus a threshold to be compared with the different diagnoses' scores. On the other hand, *therapeutic decisions* are based on a pre-defined set of parameters: effectiveness, cost, side-effects, compliance, duration. Finally, *conclusions* represent the output of a decision process (for instance, assuming a given diagnostic hypothesis is a typical conclusion of a diagnostic decision action). **Composite actions** are defined in terms of their components, via the "has-part" relation (this supports for top-down refinement in the description of GL). *Control relations* establish which actions might be executed next and in what order. We distinguish among at least four different control relations: *sequence, constrained, alternative* and *repetition*. In particular, repetition might require complex descriptions, possibly including maximum and minimum duration, frequency, periodicity, exit conditions. The description of sequences usually involves the definition of the minimum and maximum delay between actions. Complex temporal constraints between actions (e.g., overlaps, during) can be specified using constrained control relations.

Acquisition. GLARE's acquisition module (CG-AM in Figure 1) provides expert-physicians with a user-friendly and easy-to-use tool for acquiring GL. In order to achieve these goals, we have implemented: (i) a graphical interface, which supports primitives for drawing the control information within the GL, and ad hoc windows to acquire the internal properties of the objects; (ii) facilities for browsing the GL; (iii) an "intelligent" help and consistency checking.

Specifically, the acquisition tool provides an "intelligent" interface supporting expert physicians in the acquisition of a consistent GL, relying on different forms of consistency checking, including name and range checking, logical design criteria checks, and semantics checks concerning the consistency of temporal constraints in the GL (see Section 4 and [6]).

Execution. A dedicated module has been developed in order to support the execution of a GL on a specific patient, adopting the "agenda technique" (see [5]). In order to support physicians' decision making, the execution engine has been loosely coupled with a decision-making facility (DECIDE-HELP in Figure 1; see Section 5).

Testing. We have already tested our prototype acquisition and representation system considering different domains, including bladder cancer, reflux esophagitis, heart failure, and ischemic stroke. The acquisition of a clinical guideline using GLARE was reasonably fast (e.g., the acquisition of the guideline on heart failure required 3 days). In all the tests, our representation formalism proved to be expressive enough to cover the clinical algorithms. While such implementations have been successfully tested by some physicians, they have not been used yet in a real clinical setting.

3. Resource-based Contextualization of GL

Contextualization is an essential step to introduce a in the clinical practice. Actually, one of the most relevant obstacles to the exploitation and dissemination of GL is the gap between the generality of GL themselves (as defined, e.g., by specialists' committees) and the peculiarities of the specific contexts of application.

GLARE provides a facility to adapt GL on the basis of the resources available in a given context (see also [7]). In GLARE's representation formalism, the resources needed by each action of the GL are explicitly declared. This means that a check can be performed in order to prune the branches of the GL that cannot be executed because of resources unavailability. In fact, a GL describes a set of alternative paths among which the physicians can choose during the diagnostic/therapeutic process. Pruning non-executable alternative branches brings out with a context-dependent GL, that describes all and only those actions that respect the original meaning of the general input GL, and that can actually be executed in the given context (since the required resources are available).

GLARE contextualization algorithm is based on the notion of *legal path*. Informally, we define a *legal path* in a GL as a path from the starting action to an ending action such that all the resources needed by (the actions in) the path are available. An action belongs to a *legal path* if:

- it does not require unavailable resources and
- it is the last node of a path or, alternatively, there is at least one action that follows it which belongs to a legal path.

Intuitively, the algorithm we have devised navigates the hierarchical graph formed by GLARE's actions looking for legal paths, and substituting with a proper warning signal paths that are not legal (see [7] for more details).

4. Representing and Reasoning with Temporal Constraints in GL

Temporal constraints play a fundamental role in GL. For example, temporal indeterminacy, constraints about duration, delays between actions and periodic repetitions of actions are essential in order to cope with clinical therapies. Furthermore, it is also necessary to carefully take into account the (implicit) temporal constraints derived from the hierarchical decomposition of actions into their components and from the control flow of actions in the GL. Moreover, also the (constraints describing the) execution times of actions concerning a specific GL execution might be taken into account. In summary, to cope with temporal constraints in GL, the following features have to be supported:

1. qualitative and quantitative constraints, as well as repeated/periodic events; all types of constraints may be imprecise and/or partially defined;
2. a structured representation of complex events (in terms of part-of relations), to deal with structured descriptions of the domain knowledge;
3. the distinction between classes of actions (e.g. an action in a general GL) and instances of such actions (e.g., the specific execution of an action in a GL);
4. the consistency of the temporal constraints between classes and instances. This involves dealing with the inheritance of constraints (from classes to instances) and with the predictive role of constraints between classes.

Obviously, the interplay between issues 1-4 needs to be dealt with, too.

A complete automatic treatment of temporal constraints involves – besides the design of an expressive representation **formalism** – also the development of correct and complete temporal reasoning **algorithms** operating on them. Moreover, the *trade-off* between *expressiveness* and *computational complexity* must be considered, if one aims to develop *polynomial-time* and *complete* algorithms [8]. Few works in the area of GL have deeply analyzed this topic so far; furthermore, despite the large amount of works in the AI temporal reasoning community (see, e.g., the survey in [8]), no approach covers all the phenomena (1-4) above.

4.1. Temporal Reasoning in GLARE

We have provided a high-level language to represent the different types of temporal constraints discussed above. In order to devise proper and efficient temporal reasoning algorithms, we have also provided an internal representation formalism, into which the high-level temporal formalism can be easily mapped (in *linear time*). Specifically, we introduced the notion of STP-tree [9], which extends "standard" STP (Simple Temporal Formalism [10]) to cope also with repeated and periodic actions. The temporal reasoning algorithms we have devised operate on such an extended internal formalism, and propagate the temporal constraints to check their consistency. We have defined two basic temporal-reasoning algorithms:

1. *STP_tree_consistency*, that checks the consistency of the constraints in a STP-tree;
2. *IntegratedConsistency*, that, given a consistent STP-tree (representing the constraints in a GL) and an additional STP (representing the time of the executed actions, when the GL has been applied to one or more specific patients), propagates the constraints in order to check whether the execution-times are consistent with the temporal constraints in the GL.

Both algorithms are correct and complete, and operate in polynomial time on the number of actions [9].

4.2. Temporal Reasoning for Clinical GL: Applications and Advantages

During **acquisition**, an automated ***consistency-checking*** facility is important to check the temporal consistency of the GL. Such a facility can be provided through a call to the *STP_tree_consistency* algorithm, which can be advocated at any stage during the acquisition of a GL, so that incremental consistency checking is also possible.

During **execution**, our *IntegratedConsistency* algorithm can be used as the core of a user-oriented approach providing temporal facilities to user-physicians, as follows:

a) For scheduling purposes, it may be used to provide a facility to assess when the next actions have to be performed, given the constraints in the whole GL and given the time when the last actions in the GL have been executed.
b) From the point of view of quality evaluation/assessment, it may be used to provide a facility to check whether the temporal constraints in the GL have been respected or not by the instances of actions that have been executed (considering also partial – i.e., ongoing – executions).
c) In order to support decision making, it might be used in order to provide a (temporal) query-answering facility.
d) Still considering decision making, temporal reasoning can be profitably coupled with “simulation” computer-based facilities to see the temporal consequences of choosing among different alternative paths in a GL.

5. Decision Making Support

Decision making is a central issue in clinical practice. In particular, supporting therapy selection is a critical objective to be achieved. Consider that, when executing a GL on a given patient, a physician can be faced with a choice among different therapeutic alternatives, and identifying the most suitable one is often not straightforward. Actually in several situations no alternative is really “better” than the others, from a strictly clinical viewpoint, and GL are only meant to present all the range of choices, leaving to the user the responsibility of selecting the “right” one.

In clinical practice, various selection parameters (such as the costs and effectiveness of the different procedures) are sometimes available when executing a GL, but the task of comparing and balancing them is typically left to the physician. A system able to automatize the comparison and to provide quantitative results would be of great help in several real world situations.

Decision theory seems a natural candidate as a methodology for affording this analysis; to this hand, we have realized a mapping between the GL representation primitives, and decision theory concepts (see [11] for details), and we are implementing a decision theory tool for supporting therapy selection in GLARE.

In short, in a well-formed GL, a decision action is always preceded by a query action (see Section 2), that is adopted to collect all the patient’s parameters necessary (and sufficient) for taking the decision itself (here we refer in particular to therapeutic decisions: in the GL context, as a matter of fact, a diagnostic decision only allows to classify the disease the patient is suffering from, and simply preludes to a therapeutic decision among suitable alternatives to care the patient herself). Each decision is therefore based on an (explicit or implicit) data collection completed at decision time, and does not depend on the previous history of the patient (i.e. on previous data collections and on previous decisions found along the path that leads to the decision at hand). We can thus say that the GL describes a first-order Markov model, since each time a query action is implemented, the patient's situation is completely re-assessed, and an (explicit or implicit) query action is always found before a decision action. This observation justifies the mapping of GL primitives to the field of decision theory, and in particular allows us to represent a GL as a Markov Decision Process (MDP), which has been recognized as a basic representation framework for dynamic decision making under uncertainty. Note that Markov models have been widely used in the last decades in medical decision making, and represent nowadays a well-understood instrument to cope with time-dependent medical decision problems [12].

In particular, it is straightforward to define the concept of state as the set of patient's parameters that are normally measured for taking decisions and for assessing therapy outcomes. On the other hand, we consider work actions as the means to produce state transitions, since they are the only actions with a potential effect on the state variables. The utility of reaching a state can be evaluated in terms of life expectancy, corrected by Quality Adjusted Life Year (QALYs) [13]. We can derive the utility of a state from the medical literature, as we do for obtaining the probability of state transitions. Note that, for those medical fields in which the medical literature does not provide these numbers, it is reasonable to expect this information to be available in the near future. As a matter of fact, the increasing exploitation of Hospital Information Systems and of computerized GL management tools will allow for the collection of large amounts of clinical practice data, on which it will be easy to draw statistics, at least at the local level. Costs can be interpreted as monetary expenses: each work action will have a price. Additionally, costs can also be evaluated in terms of the time and the resources required to complete work actions (see [14]).

On the basis of these preliminary mapping considerations, which are general enough to be easily applied within any of the systems for the computerized management of GL described in the literature, we are implementing a decision theory facility in GLARE. In our approach, the MDP describing the GL is represented resorting to a dynamic decision network [15], a choice that allows one to explicitly take advantage of conditional independencies from the modeling viewpoint. In particular, an earlier version of GLARE already embedded a facility able to calculate costs, time and resources required to complete paths in a GL (see Section 4; details can be found in [14]); the decision theory support can be seen as an extension of that work. In particular, the GLARE decision theory facility will enable the user: (1) to identify the optimal policy, and (2) to calculate the expected utility along a path, by exploiting classical powerful dynamic programming algorithms.

6. GL Verification through Model-checking Techniques

The general methodology we propose is to exploit the capabilities of a model checker by loosely integrating with it a GL system (see Figure 1). Such a loose integration can be provided by defining a module for the automatic translation of any GL into the corresponding GL represented in the model-checker format. The translation of GL can be performed once-and-for-all a priori. After that, the model-checker can be used in a standard way in order to verify any property (that can be expressed in the model-checker property language) on any of the GL. It is important to stress that the properties need not to be defined a-priori: the user can directly express a new property and ask the model-checker to verify it.

On the other hand, in many current approaches, a specific "ad-hoc" software module is used in order to check a class of properties about GL. In such approaches the user directly invokes the specialized module which has been devised in order to verify the property s\he is interested in. In such a case, the properties to be checked must be fixed a-priori, and an ad-hoc module must have been devised for each one of them.

Our methodology is advantageous with respect to "ad-hoc" approaches, since:

1. it does not require the development of many different software modules (one for each class of properties): the translator is built once and for all, and after that any property can be checked on any GL;
2. it is much more efficient: if a new class of properties needs to be verified, no intervention of programmers (to build a new software module) is required;
3. it is more flexible. In the "ad-hoc" approaches, designers are called to foresee "a-priori" which are the classes of properties to be checked, in order to let programmers produce the proper software modules. On the other hand, in our approach, users can directly check any property (provided that it can be expressed in the language of the model-checker), also in an interactive way.

6.1. Representing GLARE GL Using Promela

In order to instantiate the above methodology, we have loosely coupled GLARE [16] with the model checker SPIN [17]. SPIN is used in the verification of GL by modelling the GL and the agents interacting with it as a process in the Promela language. Promela allows a high level model of a distributed system to be defined by modelling each agent in an extended pseudo C code, including synchronization primitives and message exchange primitives. SPIN automatically generates an optimized on-the-fly verification program from the high level specification of the system and it allows the verification of correctness claims that are specified as temporal logic formulas.

The model of the system is described by specifying the following agents as Promela processes: the *Guideline* agent; the *Physician* agent; the *Outside* agent; the *Database* agent. The *Physician* agent is modelled as a nondeterministic process which interacts with the GL by evaluating the patient data, choosing among the different alternative feasible paths and deciding among the different "backtracking" alternatives in the case of action failure. The *Outside* agent, representing the outside world, provides up to date values for data (together with the time of their measurement) when they are not already available from the database. It also stores data in the database, executes work actions and reports about their success or failure. The *Database* agent models the behaviour of the patient database, allowing for data insertion and retrieval. The *Guideline* agent models the overall behaviour of the given GLARE GL. Each construct in the GL is mapped to a Promela statement or (for complex statements) to a Promela piece of code. For instance, each work action is mapped to a conditional statement in which the action preconditions are evaluated and, in the case they hold, the action is sent to the Outside agent for execution. The sequence relation and the concurrent relation are mapped to the sequence and to the parallel composition constructs in Promela. Decisions are mapped to a sequence of messages to inform the physician about the supported paths, followed by a selection construct which allows the selected path of the GL to be followed according to the physician decision.

In our implementation, the translation from a GL in GLARE to a Promela specification takes in input the GLARE description of the chosen GL and is implemented in Java (using JDOM API) using top-down recursive parsing.

6.2. Verification of GL Properties

Our approach supports the verification of different types of properties about GL:

1. **Properties concerning a GL "per se"**. One can check if the GL contains a path of actions satisfying a given set of properties (e.g., a path including actions X and Y, or a path in which no action of type X is executed);
2. **Properties of a GL in a given context.** For instance, the model checker can prove whether there is or not a therapy for a patient affected by a given disease, in the case a specific set of resources is available (not available).
3. **Properties of a GL when applied to a specific patient.** Provided that the model checker has in input all the data in the patient record, the feasibility of a given action or path of actions on the specific patient can be proved.
4. **Integrated verifications.** Given the flexibility of the model checker, any combination of the above types of verification is feasible.

7. Conclusions

In the latest years, many approaches agreed that providing a semi-automatic treatment of GL is very advantageous, and that AI techniques can be fruitfully applied to achieve such a goal. In this paper we described the case of GLARE, showing its advanced AI features to cope with (i) automatic resource-based adaptation of GL, (ii) representation and reasoning about temporal constraints in GL, (iii) decision making support, (iv) and model-based verification.

It is worth remarking that GLARE is not the only project in which (at least some of) the issues (i-iv) above have been faced. For instance, among the most relevant related approaches we mention [18] as regards issue (i), [19, 20] as regards (ii), [21] as regards (iii), [22, 23, 24] as regards (iv). A technical analysis of the innovative contributions of GLARE's approach with respect to such approaches in the literature is outside the goals of the present overview, and can be found in the referenced publications concerning GLARE.

We think that such approaches, together with GLARE's one and many others in the GL area we couldn't mention for space constraints, substantiate our belief that AI techniques, properly adapted and\or extended, can provide crucial advantages to the (semi-) automatic treatment of GLs.

Acknowledgements

From 1997 to 2005 the GLARE project has been partially supported by grants from Azienda Ospedaliera San Giovanni Battista in Turin. Since 2005 financial support has been partially provided by Koinè Sistemi S.r.l, Turin.

References

[1] Health Telematics for Clinical Guidelines and Protocols, eds. C. Gordon and J.P. Christensen IOS Press, Amsterdam, 1995.

[2] Special Issue on Workflow Management and Clinical Guidelines, D.B. Fridsma (Guest ed.), JAMIA, 22(1) (2001), 1-80.

[3] S. Miksch K. Kaiser and S.W. Tu, eds. Proceedings of the Symposium on Computerized Guidelines and Protocols. IOS Press, Amsterdam, 2004.

[4] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E.H. Shortliffe, and M. Stefanelli. Comparing computer-interpretable GL models: a case-study approach, Journal of the American Medical Informatics Association, 10 (2004), 52–68,

[5] P. Terenziani, G. Molino, M. Torchio, A Modular Approach for Representing and Executing Clinical Guidelines. Artificial Intelligence in Medicine 23 (2001), 249-276.

[6] P. Terenziani, L. Anselma, A. Bottrighi, L. Giordano, S. Montani, Automatic Checking of the Correctness of Clinical Guidelines in GLARE, Proc Medinfo 2007 (2007), to appear,

[7] P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G. Molino, G. Correndo, A context-adaptable approach to clinical guidelines. Proc. MEDINFO'04, M. Fieschi et al. (eds), Amsterdam, IOS Press, (2004) 169-173.

[8] L. Vila, A Survey on Temporal Reasoning in Artificial Intelligence, AI Communications, 7(1) (1994) 4-28.

[9] L. Anselma, P. Terenziani, S. Montani, A. Bottrighi, Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines, Artificial Intelligence in Medicine 38 (2006), 171-195.

[10] R. Dechter, I. Meiri, J. Pearl, Temporal Constraint Networks, Artificial Intelligence 49 (1991), 61-95.

[11] S. Montani and P. Terenziani, Exploiting decision theory concepts within clinical guideline systems: towards a general approach, International Journal of Intelligent System 21 (2006), 585-599,.

[12] F.A. Sonnenberg, J.R. Beck. Markov models in medical decision making: a practical guide. Medical Decision Making 13 (1993), 332-338.

[13] M.R. Gold, J. E. Siegel, L. B. Russell, M. C. Weinstein. Cost-Effectiveness in Health and Medicine. Oxford University Press, New York, 1996.

[14] P. Terenziani, S. Montani, C. Carlini, G. Molino, M. Torchio, Supporting physicians in taking decisions in Clinical Guidelines: the GLARE's "what if" facility. Journal of the American Association of Medical Informatics (JAMIA), Proc. Annual Fall Symposium, 2002.

[15] Tatman, R. Shachter. Dynamic programming and influence diagrams. IEEE Transactions on Systems, Man and Cybernetics 20 (1990), 365-379.

[16] L.Giordano, P. Terenziani, A. Bottrighi, S.Montani, L.Donzella, Model Checking for Clinical Guidelines: an Agent-based Approach. Proc. AMIA'06, Washington, November 2006.

[17] G.J.Holzmann, The SPIN Model Checker. Primer and Reference Manual, Addison-Wesley, 2003

[18] A.A. Boxwala, Applying axiomatic design methodology to create guidelines that are locally adaptable, In Proceedings American Medical Informatics Annual Symposium 2002

[19] Y. Shahar, A Framework for Knowledge-Based Temporal Abstraction in Clinical Domains, Artificial Intelligence 90(1) (1997), 79-133.

[20] G. Duftschmid, S. Miksch, W. Gall, Verification of temporal scheduling constraints in clinical practice guidelines. Artificial Intelligence in Medicine 25(2) 2002, 93-121.

[21] S. Quaglini, M. Stefanelli, A. Cavallini, G. Micieli, C. Fassino, C. Mossa, GL-based careflow systems, Artificial Inteligence in Medicine 20(1) (2000), 5-22,

[22] M.Marcos, M. Balser, A. ten Teije, F. van Harmelen, and C. Duelli, Experiences in the formalisation and verification of medical protocols, in Proc. Artificial Intelligence in Medicine in Europe (AIME) 2003, LNCS 2780, eds., S. Quaglini, P. Barahona, and S. Andreassen, Springer-Verlag, Berlin (2003), 132-141..

[23] M. Peleg, S. Tu, A. Manindroo, R.B. Altman, Modeling and analyzing biomedical processes using workflow/petri net models and tools Proc. Medinfo'04, M. Fieschi et al. (eds), Amsterdam, IOS Press, (2004), 74–78

[24] A. ten Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen, P. Lucas, S. Miksch, W. Reif, K. Rosenbrand, A. Seyfarng, `Improving medical protocols by formal methods', Artificial Intelligence in Medicine, **36**, 193.209, (2006).

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

Glossary

AGREE Instrument: A set of criteria for assessing the quality of clinical practice guidelines, developed and validated by an international group of researchers and guideline developers (the AGREE collaboration [2]).[1]

Caregiver: A person who takes care of a patient, in general a patient's family member.

Clinical Decision Support Systems: Computer systems that support patient-specific risk assessment, diagnosis and treatment planning decisions, encouraging compliance with evidence-based or other standards of best practice.

Clinical Indicators: Clinical indicators give an indication of the quality of the patient care delivered. Indicators can be divided in structure, process and outcome indicators and 'internal' and 'external' indicators. The goal of internal indicators is monitoring and improving care processes or professional performance within the own organisation. External indicators are used for being accountable to government institutions, health insurance companies or consumers about the quality of care.

Clinical Practice Guidelines (CPGs): A standard definition is 'systematically developed statements to assist practitioners and patient decisions about appropriate health care for specific circumstances.'[4]

Completeness: An attribute of a logical system that is so constituted that a contradiction arises if any proposition is introduced that cannot be derived from the axioms of the system.

Compliance: Acting according to a guideline or a guideline recommendation.

Computer Interpretable Guidelines (CIGs): Formal representations of CPGs that can be used to provide active support for improved effectiveness and safety of clinical practice.

Consensus-Based Guideline: The most common form of guideline developed is agreement among a group of experts. In the past consensus-based guidelines were developed, but currently is a shift to evidence-based guidelines.

Consistency: An attribute of a logical system that is so constituted that none of the propositions deducible from the axioms contradict one another.

Critiquing Tool: A computer-based facility able to spot and analyse differences between the activities performed in real life (e.g., the actions performed by a physician) and the 'ideal' ones (e.g., the ones prescribed by a guideline).

Decision Support Tool: A computer-based facility that supports decision making activities.

Economical Outcome: The result of a medical action (or set of actions) on the economical cost. Subjects incurring costs depend on the point of view of the analysis (the society as a whole, the national healthcare system, the insurance company, the patient, etc.).

[1] http://www.agreetrust.org/instrument.htm [accessed March 12, 2008]

Explicit Evidence-Based Guideline: Developed as an evidence-based guideline, '...but also projects the healthcare outcomes (benefits, harms, utilization, and costs) of the change in practice on a defined population.'

Evidence-Based Guideline: Developed after the systematic retrieval and appraisal of information from the literature. 'They usually include strategies for describing the strength of the evidence, and try to clearly separate opinions from evidence ... they make statements not just about which of two treatment options is 'better', but quantify the absolute differences in outcome, including both benefits and harms.'

Formal methods: Mathematical notation and techniques for specification, development, and verification of a system.

Glyphs: They are basically composite graphical objects where different geometric and visual attributes are used to encode multidimensional data structures in combination. A simple example of a glyph is an arrow whose visual attributes length, angle, and color might be used to encode three different data attributes in a single graphical object.[2] 'A glyph is a graphical object designed to convey multiple data values.' [12]

Guidelines (Institute of Medicine (IOM) definition): A widely used definition of guidelines is that of the Institute of Medicine (IOM): 'Guidelines are systematically developed statements to assist practitioner and patient decisions about appropriate healthcare for specific clinical circumstances.' (IOM 1990)

Guidelines ('Haamstede' definition): 'A guideline is a document with recommendations and instructions to assist the medical professional and the patient in decision making, based on results of scientific research followed by discussion and expression of expert-opinions, to make effective and efficient medical practice explicit.' [11]

Guidelines (NZ Guidelines Group (NZGG) definition): 'Guidelines[3] provide guidance in decision making at each level of interaction; between health professional and consumer, between purchaser and provider, and between 'funder' and 'purchaser'.'

Guidelines Adaptation: The activity of tailoring the guideline content to the characteristics of the setting in which the guideline has to be applied. Basically, guideline flow or actions can be modified, to some extent, to better fit local resources availability and/or cultural needs. A *local adaptation* of one or more guidelines are often called (clinical) protocols.

Guideline Execution Engine: A computer program that interprets formalized clinical guidelines, exchanges information with patient information systems and performs actions towards users of these systems

Guidelines International Network (G-I-N): The Guidelines International Network[4] is a major international initiative involving guideline-developing organisations from around the world. G-I-N seeks to improve the quality of health care by promoting systematic development of clinical practice guidelines and their application into practice [6].

[2] http://www.infovis-wiki.net/index.php?title=Glyph [accessed March 12, 2008]

[3] NZGG defines also different types of guidelines (http://www.nzgg.org.nz [accessed March 12, 2008])

[4] www.g-i-n.net [accessed March 12, 2008]

Health Outcome: The result of a medical action (or set of actions) on the patient's health (survival, quality of life, etc).

Healthcare Professional: A person working in a clinical setting (physician, nurse, physiotherapist, technician, etc.).

Implementation of Guidelines: Guideline implementation involves the concrete activities and interventions undertaken to turn policies into desired results. Various implementation strategies encourage the successful uptake of guidelines and changes in clinical practice.

Information Visualization (InfoVis): InfoVis is concerned with the development of interactive visual representations of abstract, multidimensional data, information, and knowledge to help users gain a deeper understanding of the contents of a domain by revealing, for example, new insights, previously unknown facts and relationships, or providing explanations for complex situations [1]. InfoVis is the communication of abstract data through the use of interactive visual interfaces [5]. 'The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.'[1]

Interaction: 'Interaction between human and computer is at the heart of modern information visualization and for a single overriding reason: the enormous benefit that can accrue from being able to change one's view of a corpus of data. Usually that corpus is so large that no single all-inclusive view is likely to lead to insight. Those who wish to acquire insight must explore, interactively, subsets of that corpus to find their way towards the view that triggers an 'a ha!' experience.' [9][5]

Living Guidelines: Guidelines updated on a more continuous basis: living guidelines are flexible, adaptable documents and present up-to-date and state-of-the-art knowledge to practitioners.

Plan Management: It involves more than specifying a problem, generating a possible solution path to reach a goal state from an initial state, and executing this solution path. Plan management includes everything from designing a particular plan or a hierarchy of plans to the real-world execution and evaluation of such plans [7].

Task Network Model (TNM): Hierarchical model of a clinical guideline plan as a network of component tasks that are enacted over time; TNMs are typically based on a standard repertoire of generic task models (e.g., plans, decisions, actions) which are configured in a specific way for particular applications.

Temporal Abstraction (TA): The inferences needed in order to extract 'high-level' temporal patters from 'low-level' time-stamped data.

Temporal Constraints: Constraints on the time when facts holds and actions are executed; they can be roughly divided into qualitative constraints (e.g., constraints on the relative order of two actions: action A before action B) and quantitative constraints (i.e., constraints referring to the metric of time; e.g., action A lasts 5 minutes)

Temporal Constraint Propagation: The inferences needed in order to infer implicit constraint from a given (explicit) set of temporal constraints.

Temporal Data: Refers to data, where changes over time or temporal aspects play a central role or are of interest. 'Temporal data is data that varies over time.' [3]

[5] http://www.infovis-wiki.net/index.php?title=Interaction [accessed March 12, 2008]

Temporal Indeterminacy: Indeterminacy on the time when facts holds and actions are executed; Temporal indeterminacy occurs whenever the time when a fact holds (an action occurs) is not exactly specified, at the lower level of granularity supported by the system.

The GuideLine Implementability Appraisal (GLIA): GLIA is a tool for appraisal of implementability of clinical guidelines, an instrument to identify obstacles to guideline implementation.[6]

Timeline: 'A timeline is a graphical or textual display of events in chronological order and is the most used technique for interacting with time-linear visual information. It also allows the user to explore relationships among historical events.' [8][7]

Verification: Proving or disproving correctness of systems (seen as guidelines and protocols).

Visual Exploration: 'The aim pursued with visual exploration is to give an overview of the data and to allow users to interactively browse through different portions of the data. In this scenario users have no or only vague hypotheses about the data; their aim is to find some. In this sense, visual exploration can be understood as an undirected search for relevant information within the data. To support users in the search process, a high degree of interactivity must be a key feature of visual exploration techniques.' [10]

Visual Mapping: It is a mapping between data aspects and visual variables, i.e., assigning specific visual characteristics to data attributes in order to facilitate visual sense-making. E.g., the data value temperature might be mapped color.

Visual Variables: They are a specified set of modifications that can be applied to objects in order to encode information. Examples are position, length, angle, slope, area, volume, or color.

Visualization: A graphical representation of data or concepts, which is either an internal construct of the mind or an external artifact supporting decision making.[8]

Workflow: 'Automation of procedures where documents, information, or tasks are passed between participants according to a defined set of rules to achieve an overall business goal [Workflow Management Coalition]'.

References

[1] Stuart Card, Jock Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers, San Francisco, 1999.

[2] AGREE Collaborative Group. Guideline Development in Europe. An International Comparison. *International Journal of Technology Assessment in Health Care*, 4(16):1039–1049, 2000.

[3] Christian S. Jensen and Richard T. Snodgrass. Temporal Data Management. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):36–44, January/February 1999.

[4] Marilyn J.Field and Kathleen N. Lohr. *Clinical Practice Guidelines: Directions for a New Program.* National Academies Press, Institute of Medicine, Washington, DC, 1990.

[5] D.A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler. Challenges in Visual Data Analysis. In *Proceedings of Information Visualization (IV 2006)*, pages 9–16, Amsterdam, NL, 2006. IEEE.

[6] http://gem.med.yale.edu/glia/ [accessed March 12, 2008]

[7] http://www.infovis-wiki.net/index.php?title=Timeline [accessed March 12, 2008]

[8] http://www.infovis-wiki.net/index.php?title=Visualization [accessed March 12, 2008]

[6] G. Ollenschläger, C. Marshall, S. Qureshi, K. Rosenbrand, J. Burgers, M. Mäkelä, and J. Slutsky. Guidelines International Network (G-I-N). Improving the Quality of Health Care: Using International Collaboration to Inform Guideline Programmes by Founding the Guidelines International Network (G-I-N). *Quality and Saftey in Health Care*, 6(13):455–460, 2004.

[7] Martha E. Pollack and John F. Horty. There's More to Life than Making Plans: Plan Management in Dynamic, Multi-agent Environements. *AI Magazine*, 4(20):71–84, 1999.

[8] Sonia Fernandes Silva and Tiziana Catarci. Visualization of linear time-oriented data: a survey. In *Proceedings of the 1st International Conference on Web Information Systems Engineering*, pages 310–319. IEEE, 2000.

[9] Robert Spence. *Information Visualization - Design for Interaction*. Pearson Education Limited, 2nd Edition, Essex, UK, 2007.

[10] Christian Tominski. *Event-Based Visualization for User-Centered Visual Analysis*. PhD thesis, Institute for Computer Science, Department of Computer Science and Electrical Engineering, University of Rostock, 2006.

[11] J.J.E van Everdingen, J.S. Burgers, J.A. Swinkels, T.A. van Barneveld, and J.L.M van de Klundert. *Evidence-based richtlijnontwikkeling. Een leidraad voor de praktijk.* Bohn Stafleu van Loghem, Houten, The Netherlands, 2004.

[12] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, San Francisco, CA, 2004.

This page intentionally left blank

Computer-based Medical Guidelines and Protocols: A Primer and Current Trends
A. Ten Teije et al. (Eds.)
IOS Press, 2008

Author Index

This page intentionally left blank