Series on Innovative Intelligence – Vol. 3

Internet-Based Intelligent Information Processing Systems

Editors

R. J. Howlett

N. S. Ichalkaranje

L. C. Jain

G. Tonfoni

Internet-Based Intelligent Information Processing Systems

Series on Innovative Intelligence

Editor: L. C. Jain *(University of South Australia)*

Published:

Vol. 1 Virtual Environments for Teaching and Learning
(*eds. L. C. Jain, R. J. Howlett, N. S. Ichalkaranje & G. Tonfoni*)

Vol. 2 Advances in Intelligent Systems for Defence
(*eds. L. C. Jain, N. S. Ichalkaranje & G. Tonfoni*)

Forthcoming Titles:

Neural Networks for Intelligent Signal Processing
(A. Zaknich)

Complex-Valued Neural Networks: Theories and Applications
(ed. A. Hirose)

Biology and Logic-Based Applied Machine Intelligence: Theory and Applications
(A. Konar & L. C. Jain)

Levels of Evolutionary Adaptation for Fuzzy Agents
(G. Resconi & L. C. Jain)

Series on Innovative Intelligence – Vol. 3

Internet-Based Intelligent Information Processing Systems

Editors

R. J. Howlett
Brighton University, UK

N. S. Ichalkaranje
University of South Australia

L. C. Jain
University of South Australia

G. Tonfoni
The George Washington University, USA

World Scientific
New Jersey • London • Singapore • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: Suite 202, 1060 Main Street, River Edge, NJ 07661
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

INTERNET-BASED INTELLIGENT INFORMATION PROCESSING SYSTEMS

ISBN 981-238-281-X

This book is printed on acid-free paper.

Printed in Singapore by Mainland Press

Foreword

This book on Internet-Based Intelligent Information Processing Systems offers a broad survey of various topics, all related to Internet technology. It also shows the research led in this field all over the planet as the authors of these several chapters come not only from different subfields but also from different parts of the world.

What is interesting to see is that in Asia, Oceania, in Europe as well as in America, all share common interests and same methodology of research. More than ever, knowledge in the scientific area has become global. Actually, there is no significant difference between these many researchers. The quality and the interests are pretty much on the same level.

This book gives to everyone a very good and broad overview of the field of Internet-Based Information Processing Systems and I can recommend this book to anyone, but particularly to new researchers in the field. They will be able to answer the first PhD questions about who has done what, when and how. This book should be a first book to read to know the state of the art of the research in this area.

Prof. Dr. Nadia Magnenat-Thalmann

Director of Miralab
University of Geneva
Switzerland

This page is intentionally left blank

Preface

Internet-based Intelligent Information Processing Systems deals with the question of how and to what extent developments may be used to cope with major challenges, opportunities and problems arising from the advent of the Internet. Factors which form part of the development include, advanced theories and models arising from artificial intelligence, computer science and cognitive science.

It is fair to say that these observations are not primarily about the advent of the Internet which originally had a restricted and very specialized use. The following observations are about the worldwide expansion of the Internet and the Web which occurred later. It would require a complete book to provide an adequate historical background of the Internet and the Web. It would be necessary to account for the worldwide diffusion and to deal with options, doubts, decisions why this should have occurred.

There is fortunately an already wide literature dealing with each of the many aspects involved. The domains of knowledge are continuously increasing both in number and in complexity. These multiple and continuously shifting scenarios make it difficult to identify and consolidate stable knowledge in this area as each new experience necessitates revision and upgrading of existing information and knowledge.

After less than two decades of interest and expansion in the Internet and websites design there is perhaps a need to preserve old sites which are meaningful to the extent that they help provide an understanding of past development. In the accelerated time scale of high tech development, ten years ago may appear to be prehistory. Previously accumulated experience and knowledge are worth preserving as they may contain valuable experience.

Web designers of today are likely to start from scratch, and this can result in a random brainstorming operation, based upon patching together single pieces of information, which may have no depth as the originating search path and context may be lost. This may not always be the best way to proceed. Large quantities of data indicate that shortcuts should be taken if it is necessary to operate in real time. It is usually necessary to seek more efficient ways of handling excessive data if a practical system is to be developed.

Methods of simplifying and labeling information consistently and correctly are not obvious processes. Those operations require time, competence, experience and specific skills. As there exists much literature on the nature of major perceptual changes which have occurred it is not necessary to focus on this here. There is however a need to access this literature and to analyze it carefully to become aware of the significant changes which have occurred. New behavioral patterns may be observed and these need to be further considered. This literature exists in a number of forms such as case studies, specialized papers and reports. These sources often indicate existing discrepancies and a diversity of attitudes toward new information resources. These attitudes depend both on the differing arrangements and on the different cultures.

It is evident from an analysis of single cases that despite the worldwide access to information resources this does not necessarily result in the best deductions and conclusions. Although a vast amount of information is available and globally shared it may not be accurate and as such does not improve any derived conclusions. Words are meant to define and redefine reality and they are fundamental in the initial creation and shaping of such ideas. At some point words are used in an attempt to describe what occurs and to develop an understanding of the process. For example, choose 'Network' as the sample word. Network has many definitions in the New Oxford Dictionary. One definition is 'a piece of work having the form or construction of a net; a collection, arrangement, or structure with intersecting lines and interstices resembling those of a net — a chain or system of interconnected or intercommunicating immaterial things, points or people. Also a representation of interconnected events, processes, etc. used in the study of work efficiency.' Connections and exchanges are features, and these words add value to the definition. It is a fact however that those words mean different kinds of things to different individuals. This is a property of words that depend on the context to have a variety of quite different meanings.

Further, even if the meaning of the word appears to be consensually agreed upon, it is still possible that the meaning may be equivocal. We may observe that there are many different meanings triggered by this sample word. The word 'network' materialized into many different 'worlds' depending on the interpretation. It is possible to obtain conflicting interpretations of this apparently simple example.

The purpose of the World Wide Network is about disseminating and accessing knowledge. The vast extent of the Network gives the possibility of discovering new information territories of amazing extent. The network is about providing information on diverse topics covering all fields of human endeavor. Its very size makes new methods of communication and outreach available in previously unthinkable ways. Naturally the subject has already spawned a new and extended literature.

The Internet, Web or network, may be thought of in a number of different but coexisting forms. The Internet may be viewed as an assembly of information and knowledge which forms part of a wider texture. Just as a tapestry is made and built from a sequence of knots, so may the Internet be viewed as a highly intertwined and dynamic system for making knowledge available worldwide. We may think of Internet as the implementation of an old dream where we possess the beginning of an overall universal encyclopedia, which is a repository of overall knowledge. This knowledge is consistently categorized and commonly shared. The design and planning of such a system intended to make this very old dream come true, arises from the possibility of connecting and linking all kinds of elements and disciplines. The intention was to have them all available through the Internet and Web media. The possibility of merging all existing media into one has extended the original concept of an encyclopedia so as to cover the realm of encyclomedia.

Efforts have been made to have many sources of information available on the Internet. This has been extended to include for example such diverse items as: phone conversations, television and radio programs, movies, music, books, newspapers and more. Such a broad interpretation of the network has proved to be a highly controversial development. The idea that one media would predominate and incorporate all previously existing diverse media seems to be troubled by a number of negative side effects. Much has been written which explores and documents all of these aspects.

The network has proved to be effective in presenting introductory information such as brochures to potential customers, introductory material for museums, libraries, etc.

This entire area of research and practice has materialized into a wide set of highly specialized operations, based upon skills, such as graphic design has been given the comprehensive name of 'information design'. This innovative discipline is evolving in a number of different directions. Studies are also undertaken to ensure that Web visitors do not suffer from cognitive overload. This research is based on the results of observation and analysis.

It may provide users with a unique opportunity to become 'informactive' (Tonfoni 1996). This feature is the special ability to be aware of and sensitive to the specific nature of the information of interest. It may also possess the ability to choose and interpret a continuous stream of input data. According to this viewpoint the user's role is different in this case in that they are actively involved in the process. In this instance the user is directly involved in analyzing and interpreting information, and is thus in a position to make better informed decisions.

In order to develop such a system it is necessary to plan the information system using the information sensitive criteria given by Tonfoni (1998). A development of this kind is dependent on a very special kind of information system. Future Web visitors who have their own 'cognitive search engines' will then be able to make their own choices and decisions. 'Informactive' Web visitors will not have to accept black box models of reasoning which may end in blind alleys but will be able to make their own informed choices. The reasons why some solutions were provided and disclosing which of the information sources accessed are all part of an important activity.

There is a final interpretation for the network which also involves inappropriate information that is easily available on the Web. This inappropriate material do exist and may be readily accessed. Attempts could be made to make it less accessible and less visible by the use of passwords or similar devices. Material of this kind is clearly both undesirable and unwelcome. For example it may attempt to propagate undesirable ways of thinking that lead to unacceptable behavior. Such socially unacceptable information may be promulgated by the network in an effective manner. The network is a media which has caused and continues to reveal different kinds of side effects. As indicated it is more than a tool. It may perhaps be defined as a virtual land for discovery, invention and conquest. Discovery involves finding existing facts that may contain an element of surprise. Invention involves the creation and establishment of new connections between familiar concepts. By definition, invention involves the concept of novelty to some degree. Conquest involves the exploration and mastery of some new area of information. It may be said that the individual exposed to the network today will continually experience the three elements of discovery, invention and conquest.

A brief analogy may perhaps be drawn between network explorers today and the geographical explorers of the sixteenth century. Geographic exploration and discovery in the sixteenth century, was done by explorers, motivated in part by economic factors and in part by curiosity. Curiosity was at that time strictly bound to the ability of assembling a new world of recently defined knowledge, derived from and based upon a set of disordered fragments. The Latin word 'curiosus' was complemented and often accompanied by the word 'studiosus', which meant precisely being ready and willing to learn. Listening and learning were the basic attitudes that we come across in Dupront (1997). Curiosity in the XVI century was the preliminary step in discovering the 'mirabilia', which were objects of surprise and wonder due to their substantial difference. These differences were on account of their alien nature, which were progressively disclosed and analyzed.

The curious individual was inspired by the new being an unknown. He was usually someone willing to enjoy entirely new scenarios. These were often quite novel and contained particularly surprising items for observation and scrutiny. Curiosity could also be interpreted and described as that kind of pleasure derived from finding out about unfamiliar surprising things. Curiosity was then likely to evolve into a more mundane approach which required self discipline and true dedication. Curious individuals were likely to become collectors, accumulating various items considered to be of very peculiar nature. These were then analyzed, classified and stored. At that point, an individual 'curiosus' turned into an individual 'studiosus' — to someone dedicated to accurate analysis and to making sense of new and surprising information. An individual 'studiosus' adopted a scientific attitude toward further discoveries. These included direct experience and possibly travel to those marvelous new territories.

By experiencing directly new and different models of thinking in other parts of the world the individual explorer invents new models, and attempts to make sense out of the peculiar and unfamiliar events, phenomena and objects found. The explorers would then try to establish links and connections between the well known familiar reality and the new information. Many of their descriptions were based upon contrasting, comparing, drawing analogies, and alternatively, upon illustrating and acknowledging irreconcilable differences. A wide amount of further interpretive work, at various levels of depth and breadth, was specifically carried out to finally produce highly comprehensive models. The individual's wonder, while realizing how much more in fact had to be discovered, would trigger a productive sense of inadequacy meant to stimulate further search.

Missionaries of the past were most involved in geographical explorations. Reading and analyzing their diaries may help us today in finding out and then describing attitudes, reactions and overall behavior. The most interesting aspect is really about their ability to cope and to acknowledge explicitly those different still very articulated models of operation and knowledge they happened to find. Some diaries, reporting on new findings, obviously reflect astonishment and even admiration for alien ways of thinking and operating found and accurately observed and reported. Those writings certainly illustrate that explorers, meant to turn into conquerors of the new land, had not only allowed themselves the opportunity to listen and learn, but had also certainly evolved towards a genuine sensitivity that was in fact contradictory and therefore worth spending time analyzing (Tonfoni 1988).

Although discovery in the sixteenth century was primarily motivated by economic considerations, it also resulted in a number of side effects. One of these was the discovery by the European explorers of other highly developed civilizations. This new perspective ultimately had the effect of stimulating analysis and scientific classification. The currently accepted model at that time was the Aristotelian model. Major efforts were made to adapt this model to accommodate the resulting new patterns of thinking. Much patience was required to develop satisfactory definitions, but a new vision was required to accommodate the new knowledge consistently. A collaborative effort was central to this work as the discoveries of the several explorers had to be accommodated within the framework of any new model. It was necessary to develop and extend the model when new facts became known. This is an example of the scientific method where a theory is valid until supplanted by a later more universal successor. The old theory may have some valuable reference points but the new theory gives a new perspective and vision.

A Web site may display areas of information, which are very close to each other or linked by just one keyword. The information available on the Web comes from many diverse areas of knowledge and expertise. Each of these sources will have been subjected to a series of preprocessing operations. A new form of research in this area has been designed to provide us with a description of an individual's information seeking activities. This information seeking process has been given the name 'exoinformation' (Brunk 2001). This research may permit a few categories to be identified and would then be able to respond to questions such as; "How long was a particular user involved with this search area?" and "Which steps were taken?" Here distinctions could be drawn between a single keystroke and a more detailed search of the Web. There are questions such as "How long should exoinformation remain available?" Other questions such as the sampling duration required to derive a credible comment based on the particular observation could form part of this research program. Web designers may assume an authoritative stance in their particular field. They may also consult directly with other experts in the field, for accurate decision making.

Ideally, a Web construction team should have the ability to discriminate between information which was originally reliable and derivative information which has possibly become inaccurate. Ideally a Web construction team should be able to provide an overall vision to future Web explorers so as to help them with relevant clues. These clues should indicate the expectations which the explorers may have when using this particular information area.

On the Web the principles of territorial definition and territorial control are of major importance. An indication as to the specific nature of each area of information on a website is essential and equally important as the consistent identification of keywords. Defining a territory and identifying the institutions and possibly the individuals, responsible for its design, may improve the veracity of the information. A sense of responsibility should be encouraged in the world of the Web design. As in any design it is necessary to decide the duties of the device before undertaking any construction. This is just as true for Web design. Web designers must be given their proper place in the delivery of information. They must be held responsible for the success or failure of the resulting device. Recognition of this responsibility is essential. These observations are based on experience gained over a period of time. It was not always intuitively obvious in the early stage of design.

It is considered that Web design is entering the second era. The initial era which was characterized by unconstrained expansion has been completed. Today innovation is mainly based upon an analysis of the added value of a site. In order to really make this kind of evaluation an interpretation of what is already available is needed. Analysis of which criteria have been prioritized in the past must be assessed. Aspects such as the accuracy of the information which a site is able to provide and of the methods of verification must be considered at this time. It is also necessary to evaluate the data available in a similar manner in which one would evaluate the quality of an encyclopedia. Aspects which would be considered when assessing an encyclopedia or a database would be: Authority, Subject Coverage and Emphasis, Structure, Access, Accuracy, Currency, Update, Style Consistency, Relative Brevity, Ease in Updating, Completeness, Objectivity.

The continuity of structure and consistency of data could only be ensured by a collaborative effort using the Web team for each website which though ideal is not practicable.

The information design community has developed numerous solutions, each of which is intended to provide stability of information as well as readability for the displayed texts. A new design which incorporates both architectural qualities and knowledge management skills is required (Tonfoni 1998). A website may be considered to be 'ill designed' when there is no clear plan for the display of information content. The level of treatment for the information must be accessible at each Web site and must be clearly stated by the Web designer. There are specific and consensual ways to indicate choices, both for the quantitative and the qualitative aspects of information available (Tonfoni 1996).

Websites may be seen as primary information territories where first-hand information is displayed with very little processing. There may be a few comments added, or these may be seen alternatively as secondary sources of information, based upon compilations, digests, abridgments, summaries and other forms of information condensation.

The information displayed on the Web shows a very high level of granularity as each topic and domain addressed is the result of the different choices. These choices are based upon a variety of selection criteria which materialize into very diverse styles of presentation. Some Web sites may only provide explicit directions for finding additional sources of information, or they may incorporate an annotated bibliography, or a descriptive bibliography. In other domains such as social sciences, religion and philosophy, literature and literary criticism, history and the fine arts, music and language, the emphasis is given to the primary sources rather than to the derivative literature. In this case opportunities for further discussions may be provided, as are replies to questions. These questions may remain open, and constitute matters for debate due to their often controversial nature. In domains such as science and technology the verification of the accuracy of the information on the Web is of crucial relevance. Upgrading and updating conditions are also important.

Some fast growing research areas and knowledge domains require careful scrutiny as the assessment may rapidly change over time, and present findings may be challenged. Reviews of progress made in highly specific sub domains are of major relevance. In this case subject specialists are required, rather than general reviewers.

Areas, such as science and technology, are under pressure due to the rapid growth of publications. This consideration also applies to fields such as economics, finance, business and management and engineering.

Diversity in the areas mentioned indicate the need for website design and analysis.

Diverse strategies need to be identified and mastered for consistent packaging and displaying of non homogeneous information. Rhetorics (Dupriez 1991), may provide a dictionary of literary devices, which may help to sort information and present it in ways transparent to the users. Notions from literature and literary criticism may be extended to describe Web designing processes. The Web may also be thought of and conceived as a definitely high intertextual space (Allen 2000).

The relevance theory (Sperber and Wilson 1995) may provide an explanation of the gap existing between textual patterns and their intended meaning. Potential versus effective aesthetic effects may also be explored. This consideration may also provide a useful guide concerning the choice of information to be included when considering the Web architecture.

Today the word 'aesthetics' is commonly used in a very restricted sense. The word comes from Greek and indicates that aesthetics is generally about perception, and therefore concerns the perceptual impact of the webpage. A comprehensive aesthetical consideration for each webpage and website is not only about its graphic dimension and decorative aspect but also about planned and unplanned perceptual effects and side effects. A definition of the basic requirements of the Web visitors is important. It should be indicated explicitly to casual visitors what they may expect from a certain website.

Metaphors and analogies are useful devices and they may assist Web navigators in their search. In addition they are also effective facilitating devices for teaching, as borne out by literature in the field.

In order to be effective, specific rules need to be followed to ensure that target and source analogs fulfill their allotted roles (Holyoak and Thagard 1997). The target concept is introduced to establish the main idea. By using these rules and observations in the area of Web design, it is necessary to make sure that the target concept is sufficiently accessible to facilitate the recall of the source analog. The distinctive features, which are destined to remain separate, must not generate interferences as these may prevent the intended meaning from becoming visible.

It may be seen that Web design and architecture are an art and a science. This is a preliminary step toward further consideration and an in depth analysis of what a search mechanism involves. ‘Data Mining Devices’ may be conceived as single components of a more complex ‘Digging Mechanism’, meant to facilitate a search by individuals, who must be in charge of their individual search process. They need to be aware of their individual priorities and choices.

Over time we have progressively discovered a number of different ways to learn. These ways are bound to the specific nature of the subject involved and to the kind of learning required. The same idea applies to methods of data searching attitudes and techniques. There are many ways of searching and the choice will depend on the topic to be searched and to the variety of available searching techniques. One search mechanism could not cope with the variety of possibilities. Some of these possibilities have been observed and analyzed. It is likely that other techniques will be discovered in the future. Searching attitudes are likely to evolve over time so that individuals may readily adjust the search, according to their own needs. Search can be an adventurous process. This is because search is ‘about how to get somewhere’. It is not about ‘getting somewhere’ fast and directly by a known route.

It is sometimes assumed that there is a specific virtual space to be reached as fast as possible without detours. This contradicts the concept of a network, where seekers of information turn into explorers open to acquisition of knowledge and interested in obtaining information. There must be limits and constraints. These are the choices made by individuals. These are not restrictions imposed by an external filtering device which may not show the criteria used in making the choice. An external device which filters out ‘irrelevant’ information will also probably eliminate potentially useful options. Such filters need to be applied with care when designing Internet-based intelligent information processing systems. What should be the aim when designing a support system for a number of Web users? Probably the ability to provide useful clues of consistent value for the websites of interest is a good guiding rule. The capacity for accessing websites may be described in terms of "Navigational ability". "Navigation" through the Web implies an orderly search. A more superficial exploration may be described as "surfing". Other new metaphors which may be found are "mining" and "drilling". These operations are normally undertaken by the use of various "search engines".

A site may be much more efficiently explored if a clear indication of its mission is available together with an explicit indication of the preprocessing the information has undergone. Indications of the individuals or institutions who may be contacted for further help can also be useful. This recognition can be helpful to the authors and the users. Indications as to what information and the quality of the information contained may also be provided. Such clues may be of value in that they provide Web visitors substantial clues as to whether they should investigate further or to proceed to another site. All of these considerations can be useful in making decisions concerning the choice of search engines and search optimization.

The following chapters contain information on research undertaken in the various fields. An indication as to the content of each chapter is given below.

Chapter 1 provides the reader with information on techniques used for searching the network. There has recently been an interest in building distributed, unstructured, *ad hoc* networks. Some of these networks may rely on community efforts of variable unknown standards and on unreliable computing resources. There is therefore a need to provide as far as possible some method of classification. Different types of search are identified, analyzed and illustrated. The advantages and disadvantages of recent developments in search techniques for distributed networks are presented. It is indicated that the best distributed search solution is dependent on the particular application, the resources available and the characteristics of the network and any business or legal constraints.

In Chapter 2 a discussion of adaptive content mapping for Internet navigation is given. Text based documents still dominate although all kinds of information carriers are available. The advantages and disadvantages of these are illustrated. Standard methods of information retrieval are discussed and explained in detail.

A review of the state of the art information retrieval concepts and related algorithms is given. The problem of document search on the Internet and an architecture for an adaptive Internet document navigation system is also presented. Adaptive Internet navigation provides many new areas for content oriented search. The adaptive plasticity in the data structures also poses new challenges. It is hoped the approach presented will provide new insights into the question of balancing stability versus the plasticity of data structures.

Chapter 3 provides documented illustrations of the concept of flexible queries to XML (eXtensible Markup Language) information. This chapter carries an introduction to XML and to its main features. Issues such as those of XML query languages, flexible search and the role of text retrieval techniques are also addressed. In this chapter a flexible technique for searching patterns in XML datasets is given. This technique has many potential applications other than searching. Some of the applications where flexible XML searching may be profitably used are:

- brokerage where an increasing wealth of information is available;
- foreign document management, despite changes in huge amounts of XML information cross organizational borders and the document structure, commonalities are still retained.

Finally Web mining systems and vertical portals are considered. The availability of a huge information flow in XML format is an opportunity to provide high quality Web mining services using advanced reorganization and search tools specifically for XML.

Chapter 4 introduces the reader to agent-based hypermedia models such as the traditional node/link model of hypermedia. This is declared 'to have shown its limits long ago'. Early collaborative hypermedia systems are considered and referred to as 'closed architectures due to their reliance upon monolithic design and their inability to support integration and distribution of data and tasks and their lack of meaningful communication protocols.' A history of software agents is included. A description of the dynamically evolving hypermedia concept is given. Agent-based hypermedia models are explored in detail. Their features and differences are compared.

The agent-based hypermedia approach is considered to be an advance 'that adds to the basic concept of the agent as an autonomous entity that provides sophisticated behavior to the nodes and link.'

In conclusion whether they are introduced

- to enable cooperative work,
- to verify the integrity of links,
- or to implement active distributed applications,

agents share the same common characteristics: 'they are complex entities each with a state, some behavioral characteristics, some autonomy and some goals.'

Chapter 5 deals with a self organizing neural networks application applied to information organization. A preliminary clarification of document processing is defined as 'a set of techniques that includes clustering, filtering and retrieval'. A clear-cut distinction is made between the past interest in obtaining ranking algorithms for information retrieval systems or the effective information filtering algorithms and the present requirement 'for a new generation of techniques which are able to build linking relationships between documents or are able to insert the information in the right context.'

Much work has been done in the field of information retrieval. It is evident that 'document organization should be more than the ranking methods used by Web search engines. Document organization based on semantics is now becoming a central issue in information processing in order to search and browse through large document repositories.'

An introduction to document representation is followed by the description of some applications of self-organizing networks applied to document clustering and to information organization. Specific attention is given to the problem that Web pages are likely to be continually out of date due to the rapid growth of information.

Among the various applications, self-organizing networks may entail developing self-organizing maps. These may be used for organizing and visualizing information 'that help a user to build a cognitive structure that supports browsing activity. They help in the management of a large information space, especially when the user is unaware of the information contained in the space.' These cognitive structures are needed 'in order to help avoid the effects of an information overload.'

Chapter 6 gives an overview of 'emotion orientated research'. It is stated that while 'science has made great advances in the past century scientific and engineering technologies have not always dealt with human emotion explicitly. Humans who use these technologies do not necessarily exhibit reasonable behavior because humans have emotions and may be influenced by them. The interaction between emotion and cognition that treats the emotion as an obstructive thing may be viewed from a psychological point of view as being old fashioned. Since the research field of engineering aims to achieve a smooth communication between humans and machines, it is necessary to develop a technique for understanding human's emotions. This understanding is essential if it is hoped to embed emotional functions in the machines.' In this chapter a description of an emotion orientated intelligent system, with a human-like communication ability is provided.

The system is said to have functions such as the recognition of nonverbal messages such as facial expressions. It is said to recognize a specific emotion triggered by music or by watching pictures. In conclusion it is too difficult to define human emotions owing to their complexity. Our current techniques are not subtle enough and it will be necessary to develop a new method.

The description of 'a system for supporting knowledge creation in a network community is given in Chapter 7. Called a public opinion channel, it automatically creates and broadcasts radio and TV programs based on messages received from community members. The programs broadcast would include minority opinions as well as majority ones. The underlying design concept and the prototype are discussed using viewpoints from social psychology and from cognitive psychology. The concepts of 'network community' and 'community knowledge' are also explored in the introduction to this chapter. Further terminological specifications are illustrated such as 'knowledge creating community'.

This contribution shows 'that there are many questions that need to be answered if we are to understand and support a knowledge creating community. It would be necessary to know what knowledge creation is, why communication tools have difficulties supporting it, and what features and functions are needed for the tools. From the viewpoint of cognitive psychology, we propose a notion of metacognition that enables us to change our ways of thinking and to create knowledge. We have implemented a prototype system consistent with these proposals and our preliminary experiments with that system have provided us with guidelines for designing a communication tool for supporting knowledge creation in a community. We think in the future we will be able to design better network communication tools.

Chapter 8 gives a description of ‘an innovative agent-based system framework designed to facilitate the implementation of intelligent agent based e-commerce applications.’ Attention is given to the rapid development of e-commerce, where the Internet is becoming ‘a common virtual marketplace to do business, search for information and communicate with each other. Owing to the increasing amounts of information, searching, knowledge discovery and Web mining are becoming the critical key to success. Intelligent software applications are becoming a potential area of development for intelligent e-business in the new millennium.’ These software applications are also known as Agents.

It is proposed that ‘an integrated intelligent Agent based framework, known as iJADE – ‘intelligent Java Agent based Development Environment’ could be used in conjunction with a PC for this task. To accommodate the deficiency of contemporary agent software platforms such as IBM aglets and object space voyager agents, which mainly focus on multi agent mobility and communication could be utilized. iJADE provides an ingenious layer called “the conscious intelligent layer” and it supports different AI functionalities for the multi agent applications.’

In conclusion it may be said that ‘from the implementation point of view, two typical intelligent agent based e-commerce applications, namely the iJADE authenticator and the iJADE W shopper, are introduced as examples. It is hoped that this development will herald a new era of e-commerce applications which use intelligent Agent based systems.’

In Chapter 9 a description of ‘a real time, Internet-based S&P (Standard and Poores) future trading system, includes a description of the general aspects of Internet-mediated interactions with electronic exchanges. Inner-shell stochastic nonlinear models are developed, and Canonical Momenta Indicators (CMI) are derived from a fitted Lagrangian used by outer-shell trading models dependent on these indicators.

Recursive and adaptive optimization using Adaptive Simulated Annealing (ASA) is used for fitting parameters shared across these shells of dynamic and trading models.’

This contribution is about automated Internet trading based on optimized physical models of markets. Real-world problems are rarely solved in a closed algebraic form. Methods must be devised to deal with this complexity and to extract practical information in finite time. Despite this, the premise is that we can develop a robust and consistent model of markets. This is true in the field of financial engineering, where time series of various financial instruments reflect non-equilibrium, highly nonlinear, possibly even chaotic underlying processes. A further difficulty is that there is a huge amount of data to be processed. Under these circumstances, to develop models and schemes for automated profitable training is a nontrivial task.

The conclusions of this work are summarized and the authors make a statement.

‘We have presented an Internet-enabled trading system with its two components: the connection API and the computational trading engine. An important result of our work is that the ideas for our algorithms, and the use of the mathematical physics algorithms, must be supplemented by many practical considerations en route to developing a profitable trading system. Lastly it is shown that a minimal set of trading signals can generate a rich and robust set of trading rules that identify profitable domains of trading at various time scales. This is a confirmation of the hypothesis that markets are not efficient, as noted in other studies.'

Chapter 10 describes the implementation and maintenance of a case based reasoning system to support heating, ventilation and air conditioning systems, sales staff operating in remote locations. 'The system operates on the World Wide Web and uses XML as the communications protocol between the client and server Java applets. The paper describes the motivation for the system, its implementation and the trial of the system. The benefits to the company are detailed. The system's case base grew rapidly and this caused a problem of case redundancy. A simple algorithm to identify and remove redundant cases is described. The paper notes that had the maintenance of the case base been considered more carefully during the system design, some of this maintenance would have been unnecessary.

The chapters are not homogenous and indicate the chaotic nature of the Web today.

The domains identified and the problems posed are very diverse. This anthology has in fact been designed to represent the complexity of the field today. It indicates the variety of existing solutions in a realistic manner. The motivation has been to show what exists now and hope that it will in the future evolve in an orderly way and not in a chaotic manner. It is to be hoped that it will help define realistic criteria in the future.

We are indebted to all the contributors and reviewers for their excellent work. We sincerely thank Berend Jan van der Zwaag for his excellent help in the preparation of the manuscript. We hope that the readers will find this book useful.

References

Allen, G. (2000), *Intertextuality*, Routledge: London, New York.

Brunk, B. (2001), "Exoinformation and interface design," *ASIST*, vol. 27, no. 6, pp.11-13.

Dupriez, B. (1991), *A Dictionary of Literary Devices*, University of Toronto Press, Toronto.

Dupront, A. (1997), *Le Mythe de Croisade*, Gallimard, Paris.

Holyoak, K. and Thagard, P. (1997), "The analogical mind," *American Psychologist*, vol. 52, no. 1, pp. 35-44.

Sperber, D. and Wilson, D. (1995), *Relevance: Communication and Cognition*, 2nd edition, Blackwell, Oxford. (Original publication 1986.)

Tonfoni, G. (1988), "Problemi di teoria linguistica nella opera di Hervas y Panduro," *Lingua e Stile*, vol. 23, no. 3, pp. 365-381.

Tonfoni, G. (1996), *Communications Patterns and Textual Forms*, Intellect, U.K.

Tonfoni, G. (1998), *Information Design: the Knowledge Architect's Toolkit*, Scarecrowpress, U.S.

Contents

Chapter 1.
A review of search and resource discovery techniques in peer-to-peer networks
S. Botros and S. Waterhouse

Chapter 2.
Adaptive content mapping for Internet navigation
R.W. Brause and M. Ueberall

Chapter 3.
Flexible queries to XML information

E. Damiani, N. Lavarini, S. Marrara, B. Oliboni, and L. Tanca

Chapter 4.
Agent-based hypermedia models

W. Balzano, P. Ciancarini, A. Dattolo, and F. Vitali

Chapter 5.
Self-organizing neural networks application for information organization
R. Rizzo

Chapter 6.
Emotion-orientated intelligent systems
T. Ichimura, T. Yamashita, K. Mera, A. Sato, and N. Shirahama

Chapter 7.
Public opinion channel: a network-based interactive broadcasting system for supporting a knowledge-creating community
T. Fukuhara, N. Fujihara, S. Azechi, H. Kubota, and T. Nishida

Chapter 8.

A new era of intelligent e-commerce based on intelligent Java agent-based development environment (iJADE)

R.S.T. Lee

Chapter 9.
Automated Internet trading based on optimized physics models of markets

L. Ingber and R.P. Mondescu

Chapter 10.
Implementing and maintaining a Web case-based reasoning system for heating ventilation and air conditioning systems sales support

I. Watson

Chapter 1

A Review of Search and Resource Discovery Techniques in Peer-to-Peer Networks

S. Botros and S. Waterhouse

In this chapter we address the problem of searching for content and resources in a dynamic network. We review and classify the different approaches to search in a dynamic network. Specifically, we review "small-world" search strategies, content-addressable networks, and learning and self-organization strategies. Each of these strategies has its advantages and disadvantages for different applications. For example, the small world approaches are simple, but are generally less efficient at finding rare resources. Content-addressable networks on the other hand allow for the search of any resource by content in a guaranteed number of node hops, but require sending control and maintenance messages between nodes, especially if the network is too dynamic. Learning and self-adaptation strategies may improve the performance of search, but are less efficient if the network nodes are not consistent. Finally, we present JXTA Search, which is a network of hubs that index the content of a peer-to-peer network. JXTA Search uses a publish/subscribe strategy for indexing content. We propose an organization of the JXTA search hubs, which reduces the storage and communications requirements.

1 Introduction

Recently there has been a great interest in building distributed, unstructured, ad hoc networks, which are not necessarily supported by reliable servers and routers that traditionally characterize networks such as the Internet. Some of these networks may even rely on community efforts and unreliable computing resources. Examples of such efforts include peer-to-peer networks, ad hoc wireless networks, networked objects for ubiquitous computing and smart sensor networks. This recent interest has been motivated by the increased availability of communications bandwidth and the advances in communications-enabled computing devices.

The following features usually characterize these classes of networks:

- Nodes in the network can act as servers, clients and sometimes routers.
- The network topology and content is very dynamic in the sense that nodes are generally unreliable and can appear and disappear randomly. The content and services supplied by nodes are also dynamic and random.
- Nodes may or may not be restricted to which other nodes they physically connect to, but they usually maintain a set of virtual neighbors that they usually communicate with. Each node only has a limited view of the network, and its content, and usually requires the assistance of other nodes in reaching and finding resources in the network.

The dynamic nature of these networks makes it hard to track and find content or resources. For example crawler-based search systems such as Google or Altavista are not suitable for content-searching in peer-to-peer networks due to the rapid change in content and server locations.

In this chapter we will address the problem of searching for content or other resources in the general class of peer-to-peer networks. We define a peer-to-peer network as any network of, usually heterogeneous, communicating devices, which can communicate directly with each other.

There are a lot of potential consumer and business applications for peer-to-peer systems. Applications in the consumer space include file-sharing, distributed games, decentralized marketplaces, content distribution, messaging and communications. Applications in the business space include inter-business and intra-business collaboration such as workgroup collaboration, distributed databases, planning between supply chain partners, decentralized business market places, and customer sharing. There may also exist some strategic or geopolitical reasons for companies to prefer a distributed peer-to-peer solution over a more centralized approach.

This chapter is organized as follows. In Section 2, we describe the peer-to-peer distributed search problem. In Section 3, we classify the different approaches to distributed search. In Section 3, we describe the potential role of learning and self-organization in distributed search systems. In Section 4, we propose a distributed search network based on JXTAsearch, an open-source, peer-to-peer distributed search system developed by Sun Microsystems project JXTA. Finally, we conclude the chapter in Section 5 with potential future work.

2 Search in Peer-to-Peer Systems

The problem of searching for content or resources is a key problem for peer-to-peer systems. Most applications of distributed peer-to-peer networks, such as file sharing, instant-messaging, personalized event notification, and chatting, involve finding objects or resources of interest (e.g. devices, services, persons or information) or exchanging resources, such as files, with other peers. This is usually accomplished by a system of advertisements and queries. For example, resource providers may publish ("advertise") resource availability together with the resource description, and resource consumers send search queries, which express their resource needs, across the network to be matched with the published advertisements. Another alternative is for resource seekers to advertise their needs on the network and resource providers actively query the network for resource needs that they can match. There may be few other variations to the above alternatives, but they all share an advertisement/query model. In the rest of this chapter, we will refer to advertisement publishers and advertisement consumers to refer to the two types of activities involved in search. Viewed in this way, the problem of search is basically reduced to the problem of querying a dynamic and distributed directory of advertisements by advertisement consumers. This distributed directory is built using a subset of all the peers in the network.

Figure 1 shows an example of a fully decentralized, basic peer-to-peer network. Peers in the network may be any device with some networking capabilities. Not all the nodes in the network implement all the functionalities of the peer-to-peer network(e.g. server, client or router functionalities). Each node has a limited view of the overall network resources, and relies on message forwarding between nodes to get a query answered. For example node 1 is logically connected to nodes 2, 4 and 9, but does not directly know about the rest of the nodes. When a peer such as Node7 joins the network, it sends a "Join Message" to any of the nodes in the network. It also publishes the advertisements, which describe the content that this node carries. If node 1 is searching for content contributed by node 7, it propagates the query to its neighbors, which may happen to store node 7 advertisement.

The example represented in Figure 1 raises several questions that we will attempt to address in this chapter. First, is there a network organization, which improves the search efficiency? Second, what are good strategies for propagating advertisements between peers, which make finding resources easier? What is the effect of such strategies on advertisement updating? Third, what are effective query propagation strategies? What is the communication cost of these strategies and how is the query propagation strategy related to advertisement propagation? Finally, what are the technical and practical issues that govern the content of the advertisements, the design of the advertisement directory and the query propagation?

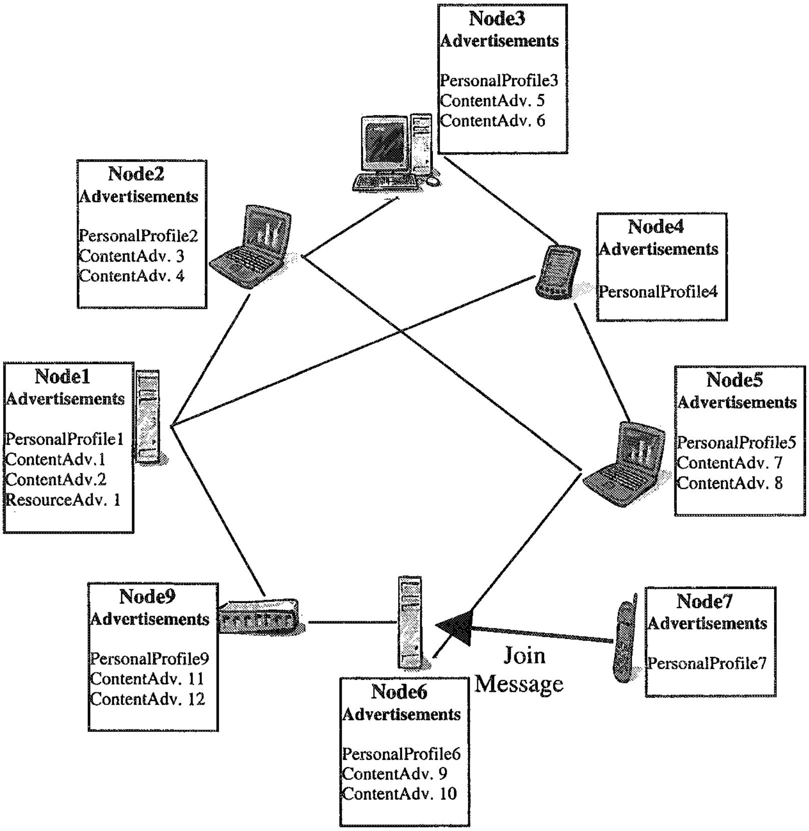


Figure 1. An example of a basic peer-to-peer network.

Technical issues include scalability of the design, maintaining the consistency of the directory, the efficiency of directory updates, the efficiency of search and the storage and communication capacities of peers. Because peer-to-peer networks tend to be very dynamic, given the random appearance and disappearance of resources, and the possibility of resources to change their physical network address from time to time, dealing with these technical issues represents a special challenge. Practical issues that govern the design of the advertisement and query system include maintaining anonymity and privacy, dealing with network spam and malicious peers, and economic and legal issues. For example, to maintain anonymity, advertisements may not directly contain the address of the source of a resource, and queries may not contain the address of the resource seeker. Instead, advertisements and queries are propagated through peers, and contain only a reference to the immediate neighboring peer from which they came.

3 Classification of Distributed Search Methods

We classify the different approaches of distributed search into two broad classes: content-agnostic search, and content-based search. These two classes differ in the network organization and in the method of advertisement and query propagation. In the content-agnostic search methods, the focus is on the organization of peers, and on the maximum distance between peers. In this class of distributed search systems, queries need to reach a subset of peers that contain a complete set of published advertisements, in order to guarantee an exhaustive search. In content-based approaches, the organization of the network, together with the query and advertisement propagation is guided by the content of the advertisements and queries. Queries in the content-based systems are usually propagated in the direction of peers that are likely to answer the queries. Both content-agnostic and content-based search methods may use learning and self-organization strategies to improve performance over time.

3.1 Content-Agnostic Search

In content-agnostic search, the organization of the peers does not depend directly on the resources they index or point to. We distinguish several types of content-agnostic distributed search networks. These types include central mediator networks, networks forming random connected graphs, and networks which have a regular structure.

In central mediator networks all peers register their content with a central server, and query that central server for all their information needs. The central mediator may act either as a matchmaker or a broker (Decker *et al.* 1996). The matchmaker approach is the approach used by Napster, the famous music-sharing program, for example. The advantages of the central mediator approach are that the search is comprehensive, the update of the directory is fast, and the total number of messages exchanged is minimized. However, the disadvantages are obvious: a central point of failure, a non-scalable solution and the requirement of a central authority. Some of these problems may be alleviated by using a decentralized network of mediators, where mediators here are simply "super-peers" who have higher level of computing and communication resources than regular peers. Later in this chapter, we propose a topology for connecting a network of JXTA search mediators.

Search networks forming random connected graphs, where nodes are connected to few random neighbors, have a relatively small distance between any two nodes. The average number of hops between nodes is approximately equal to $log(N)/log(k)$, where N is the number of nodes and k is the average node degree. The existence of a small network diameter gives only a lower limit to the number of hops between two peers, but does not necessarily imply that such a limit can be achieved using any routing algorithm. The Gnutella network (Clip2 2000) is one popular example of a random graph network. In Gnutella, each peer maintains a local index of its own content, it also maintains connections with few neighbors. By default, a peer does not advertise its content to other peers. When a peer issues a query, it broadcasts the query to all its neighbors, which in turn broadcast it to their neighbors. In a randomly connected network, with a diameter *d*, the query is expected to reach peers in *d* hops. However network traffic tend to grow with N^2. To limit the amount of network traffic, queries may have a maximum number of hops parameter, also called Time-To-Live (TTL), so only peers that are TTL hops away from the query originator will be able to respond. This distributed search approach is very flexible, and requires minimal publishing and updating of the index. When a node disappears, or changes the content it is sharing with the network, there is no extra communication cost. However, this comes at the high cost of searching the index. For an exhaustive search, queries should reach all nodes in the network. This is an example of the tradeoff between publishing and updating the index, also known as control complexity (Gibbins and Hall 2001), and searching the index measured by the query propagation cost. The basic Gnutella approach is a good approach when the content and the nodes themselves are very dynamic and therefore the index update cost will be very high, if advertisements would be propagated to other nodes. Several approaches have been proposed to reduce the search cost of the basic Gnutella network. One such approach is to introduce "super-peers" and use a hierarchical organization of peers. Super-peers are similar to the mediators described in the previous section. They store and maintain an index of all the regular peers they know. The availability of the super-peers eliminates the query propagation to the last layer of the graph, which is usually the layer with the least resources. This idea has been implemented in the clip2 reflector for example (Clip2 2000). Another improvement is to use feedback and learning together with some performance metrics to choose neighbors, and detect and drop connections that return little value, such as spammers and free riders (Adar and Huberman 2000, Peck 2001). One potential problem of using this approach is that it requires some time to build statistics using neighbor messages, and the network may be more dynamic than the time it takes to build those statistics.

3.1.1 Power Law Networks

One interesting idea to improve search in random networks, is to take advantage of the power law link distribution of naturally occurring networks (Adamic *et al.* 2001). As a result of preferential attachment to popular peers, networks tend to develop connectivity that has a power law distribution, that is few nodes will have very high connectivity and many nodes with low connectivity. The power law search algorithm proposed by Adamic *et al.* (2001) requires two modifications to the basic Gnutella approach. First, instead of broadcasting the query to all neighbors, broadcast it only to the neighbor with the highest number of connections, which has not seen the query yet. Second, nodes exchange their stored content with their first- and second-degree neighbors. This power law search algorithm significantly reduces the number of hops required to answer a query. The most significant reduction is mainly due to the advertising of the node content with their neighbors and second neighbors, because only few nodes need to be visited to examine most of the collection of advertisements, and we do not need to send the query to every node. The fact that the network has a power law distribution of the links, even performing a random walk from node to node, will result in significant reduction in the number of nodes visited. This is because a random walk will tend to select high degree nodes. However, specifically choosing high degree nodes to traverse first, improves search further.

What is interesting about the power law search algorithm described above is that the network is transformed into a network of decentralized mediators, which was described in the previous section: few nodes index the resources of a high percentage of the network, and these nodes should be traversed first. In fact, smaller nodes do not need to index their neighbors' content for the power law algorithm to work. This will eliminate a lot of the advertisement publishing and update traffic, without much degradation in the search cost.

Propagating the index of a peer to its neighbors is also an idea that has been proposed by others (Rohrs 2001, Prinkey 2001) as a means for avoiding broadcasting queries to all neighbors. However, instead of limiting the propagation to second order neighbors, the propagation of the index will continue in a summary form, up to a modifiable horizon. The authors use summaries of the index instead of the total index by using hashing functions.

3.2 Content-Based Search

In this class of peer-to-peer networks, the content of queries is used to efficiently route the messages to the most relevant peers. The peers may also be organized based on the content they index to allow fast traversal. Content-based search techniques include content-mapping networks (Ratnasamy *et al.* 2001, Dabek *et al.* 2001, Zhao *et al.* 2001, Druschel and Rowstron 2001) and some variations of publish/subscribe networks (Heimbigner 2001).

In content-mapping search networks, when a peer joins a network it is assigned a responsibility to index a "zone" of the advertisement space. This is done in such a way that the union of the indices of all peers covers the whole advertisement space. The zone assigned to each peer is dynamic and depends on the number of peers joining or leaving the network at any time. When a peer needs to publish or update some advertisement, it maps the advertisement content to a location in the advertisement space. Some efficient local routing mechanism is then used to reach the peers responsible for that part of the space to register the advertisement in their index. Similarly, when a peer issues a query, this query is first mapped to a location in the advertisement space, and the same routing mechanism is used to route the query to the peers responsible for indexing that zone. The mapping of content is usually performed using hash functions, but could also be done by segmenting content into categories and subcategories.

Recent work in content-based search include content-addressable networks CAN (Ratnasamy *et al.* 2001), Chord (Dabek *et al.* 2001), Tapestry (Zhao *et al.* 2001) and Pastry (Druschel and Rowstron 2001).

3.2.1 Content Addressable Network (CAN)

The CAN network is organized in a *d-dimensional* torus. Each peer logically occupies a zone in this d-dimensional space and knows all its neighbors. Content and queries, in the form of (key,value) pairs, are also mapped into *d* dimensions using *d* global hash functions. Routing is performed from the source to destination along the coordinate that reduces the remaining distance to destination. The authors propose several techniques to make the routing more efficient and the network more robust to failure. The basic CAN approach requires on average $(d/4)N^{(1/d)}$ hops to reach the destination, and requires a routing table at each peer of *2d* neighbors.

3.2.2 Extensions to CAN

The CAN algorithm is a clever algorithm for developing and maintaining a dynamic, decentralized index. However, it assumes that peers have somewhat similar capabilities and resources, and that they are equally reliable. In practical peer-to-peer systems, a high percentage of peers is very dynamic, and unreliable, while few are more reliable and less volatile. Moreover, peers in the basic CAN approach, only maintain knowledge of their local neighbors. We have seen in our analysis with random graph networks, that introducing few random long distance neighbors reduces the diameter of the network, without increasing size of the routing table significantly. Using multiple realities, proposed by the authors, allow peers to know more distant peers in a different reality and therefore improves routing performance. But this comes at the expense of increasing the routing table by *2d* for each additional reality. We propose here few ideas borrowed from our previous analysis of random graphs which could improve the performance of the CAN network further. First, we can introduce random long-range connections between peers, peers do not need to be neighbors in the coordinate space. Introducing random long-range connections, while keeping the local routing table results in significant savings in routing distance. For example, Kleinberg (1999) proposed using long range links with probability that is inversely proportional to the square of the lattice distance between two peers. The average path length for the Kleinberg algorithm is polylogarithmic with the size of the network. Another possibility is to use Adamic's power law search algorithm (Adamic *et al.* 2001). In this case, more powerful peers will have a bigger routing table. This could be achieved, for example, by letting more powerful peers occupy multiple virtual zones and smaller peers fewer zones. Peers need to know the number of zones their neighbors occupy and where they are. The power law algorithm was designed for the case when the target of the search is not known. However, in the case of the CAN network the target peer is known, and we adapt the power law search algorithm so that peers route queries to the neighbor that has a zone closest to the target.

The above proposed approaches may improve the average path length, but with an increased price in publishing and updating the distributed index, and dealing with node addition and removal. These variations are more useful when we have a priori knowledge of peers reliability and resources.

3.2.3 Chord

Chord is another scheme for content mapping to different peers. Similar to CAN, chord resolves a key to a specific peer using consistent hashing. In the Chord algorithm, each peer is mapped to an *m-bit* identifier. Keys are also mapped to an m-bit number using a hash function such as SHA-1. The identifier space therefore forms a one-dimensional circle, as opposed to the d-dimensional CAN network, with peers and keys distributed around the perimeter. The routing table size for an *m-bit* key has *m* entries. By using a logarithmic scaling, the routing table at each peer has more links to neighboring peers than to far away peers. Therefore each hop reduces the distance to the target in identifier space by at least one half. Therefore, the search cost in chord is logarithmic in the number of peers. The update cost of a peer joining or leaving the network is $O(log^2(N))$.

There are few optimizations to the chord algorithm, which the authors suggest. For example each peer can maintain a location table of all other peers it encountered, and how close they are in physical distance. Then when a peer is routing a message, it may choose the peer from the location table instead of the peer selected by the chord algorithm if the physical distance to that peer is closer. One other optimization, is to allow peers to have several ids. This will be useful to even out the distribution of keys to the different peers or to make more reliable and powerful peers responsible for indexing more of the content space.

In addition to CAN and chord, there are few other content-mapping networks that are based on Plaxton's algorithm (Plaxton *et al.* 1997) which was originally developed for web-caching systems. For example, a variation of the Plaxton algorithm is proposed in Tapestry (Zhao *et al.* 2001) which is used in the network storage project OceanStore. Plaxton's algorithm is also used in Pastry (Druschel and Rowstron 2001). Like chord, the Plaxton algorithm maps peers and keys into an *m-bit* one dimensional identifier. The routing table in the Plaxton algorithm tends to be larger than in chord and CAN, but the number of hops to reach a peer tends to be smaller. The insertion of a new peer in the network is also generally more expensive than either the CAN or the chord approaches.

The content addressable approaches like CAN, Chord, Tapestry and Pastry, are excellent approaches for content that can be described by simple attributes. However, these approaches are more expensive for mapping content and queries that are described by multiple attributes. For example a peer who wants to subscribe to a personalized news service may have very specific interests (e.g. interest in football and specific teams). If the peer wants to advertise its interest, it would be hard to map it to a specific location (i.e. specific peer).

3.2.4 Publish/Subscribe Networks

Publish/Subscribe networks have been proposed for event notification systems (Heimbigner 2001). It is a different approach for content routing. Peers first agree on a certain query and advertisement template. Peers subscribe their needs (e.g. what events they need to be notified of) with a publish/subscribe server in a form of filter or pattern to be matched. Publish/subscribe servers form a peer-to-peer network. Each publish/subscribe server propagates and aggregates its subscriptions to other publish/subscribe servers. Information producers publish their content with the publish/subscribe servers, and this content gets routed through several publish/subscribe servers based on the aggregated subscription filters at each publish/subscribe servers. The information finally reaches all clients with relevant subscriptions. Routing efficiency is achieved by aggregating filters to keep only the most general ones.

The publish/subscribe model has also been proposed for file-sharing networks (Heimbigner 2001), using an equivalent query/ advertise model. Clients publish their resources with the servers, using a filter pattern. The patterns are then propagated to other servers in the network and aggregated as before. Queries are routed to clients who published a relevant pattern.

There are several benefits to the publish/subscribe model. Routing is based on complex content, unlike in the content-mapping networks where routing is based on a single key. Second, the servers have some central control, which may be good for preventing malicious peers. Third, aggregating subscriptions reduce the sizes of routing tables. However, the current approach has some drawbacks. One disadvantage, for example, is that subscriptions need to be forwarded to all servers. This is obviously not scalable. Another disadvantage is that the topology of the publish/subscribe network is not explicitly specified. Any peer may have any neighbors. This may result in large number of hops between information providers and information consumers. Moreover, cycles in the publish/subscribe network graph need to be detected and eliminated.

3.3 Learning and Self-Organization Strategies

Self-organization here refers to algorithms by which a node dynamically selects and changes its neighbors based on several criteria and metrics. The metrics used in selecting neighbors could be static, such as physical proximity, or dynamic, such as changing performance feedback data resulting from node interactions. In dynamic self-organization the topology of the network evolves over time based on the accumulated experience at each node. Nodes choose which nodes to connect to, based on some metrics. The metrics chosen depend on the application, the individual resources available at the different nodes, and the use patterns at the nodes. Examples of Peer-to-Peer systems with adaptive links include Alpine (Peck 2001), Freenet (Clarke 2001) and NeuroGrid (Joseph 2001). Network adaptation has been advocated as a means to improve search results, and to reduce spam and free-riders. For example, Houston (2001) proposes one self-organization network formation strategy for a distributed file-sharing application in order to reduce "spammers" and "leechers". Houston's proposed method is based on the "egocentric" evaluation of each connection, that is the evaluation of the value of the connection from the point of view of the node. Under such an approach, we expect that problem nodes, such as spammers, leechers and nodes with bad connections, will have a hard time connecting to the rest of the high value nodes. An egocentric network formation strategy will help separate the wheat from the chaff, where the wheat and chaff are defined from each node's perspective. Using a metric that is based on the quality of search results, nodes will tend to cluster with similar peers.

A self-organization and adaptation strategy can also help a node build specific knowledge about the connections it knows. This knowledge can be used for example to influence the local content advertisement and query propagation strategies that a node makes. People use an analogous strategy in their communications in real life: overtime people develop a network of connections based on their interactions with others. This network of connections tends to cluster along people with similar interests. In addition, people develop knowledge about the attributes of their different connections and use this knowledge in making decisions about the flow of information to the different connections.

A similar approach, called adaptive social discovery has been proposed in Alpine networks (Peck 2001). In adaptive social discovery, each node builds a profile for each neighbor that it communicates with. Then from the history of interactions, a peer knows which of his neighbors is more likely to answer a particular query. In order to build such a profile, the method assumes some persistence of connections, and some response feedback mechanism to determine relevant responses from the non-relevant ones. A similar learning approach has also been proposed in the Neurogrid peer-to-peer system (Joseph 2001). NeuroGrid, however, maintains a knowledge base for all the nodes it encountered.

Self-organization strategies may not work for many distributed applications. For example, if nodes are very dynamic, and appear on the network for a short time and disappear, it is hard to gather the information needed to infer the node quality or value. Moreover, since nodes appear and disappear at random, it is hard to maintain the same circle of "friend" nodes.

3.3.1 Active Versus Passive Self-Organization

Self-organization may be passive or active. Passive self-organization only requires the monitoring of the traffic passing through the node to determine high value nodes, from the node's point of view, and then attempt to establish connections with those high-value nodes. Active self-organization requires active search, which means propagating advertisements about the node attributes and/or queries seeking nodes with specific attributes. We will discuss the subject of content-agnostic and content-based information propagation strategies used in distributed search in a later section of this chapter.

4 JXTA Search

JXTA search is a network of hubs, and a set of XML-based communication protocols. Information providers register their advertisements with the hubs. The XML-based registration messages contain a "query-space", a set of predicates and the information provider address. A query-space defines the space of queries the provider can answer. The predicates define the content that the information provider is exposing to the network. An example of a registration is shown in Figure 2.

The JXTA search hubs build an inverted index of the registration content. The information consumer peer, sends an XML-based query to a JXTA search hub, which in turn searches its local registration index and routes the query to the relevant information provider peers, or other JXTA search hubs which are more likely to answer the query.

```
<register xmlns=“http://search.jxta.org”>
  <title>JXTA Stock Quote Provider</title>
  <link>http://search.jxta.org</link>
  <description> Given a ticker symbol,
    returns a 15-minute delayed quote
  </description>
  <query-server>
    jxta://59616261646162614A757874614D5047
    CF403C5700 D44AE68F9FB626DD3F18E5401
  </query-server>
  <query-space  uri=“http://search.jxta.org/text”>
    <predicate>
      <query>
        <text>sunw aol orcl</text>
      </query>
    </predicate>
  </query-space>
</register>
```

Figure 2. An example of a JXTA Search Registration Message.

5 A Network of JXTA Search Hubs

Since any peer that implements the JXTA search service can become a hub, we need a scalable organization of hubs which can accommodate thousands of hubs. We propose here one such organization.

The JXTA search hubs are organized into *N* distinct groups. We call these groups advertisement groups. Each hub in an advertisement group maintains a local index of its registered advertisements, and a “summary” of the content of the other hubs in its advertisement group. The summary exchange is performed using a Bloom Filter approach as described by (Fan *et al.* 2001). The division into advertisement groups is done in a completely decentralized fashion, but may also be centralized with one central server assigning advertisement group membership. There are several ways a decentralized division into *N* advertisement groups may be done. One alternative may be based on some function of the server unique ID, which maps the unique ID to one of the advertisement groups. Another alternative may be content-based relying on query-spaces or content categorization.

Similarly, each JXTA Search hub is a member of a network of hubs, which has at least one representative hub from each advertisement group. We call these groups, query groups. Since each hub knows, at least in a summary form, the content of all the hubs in its advertisement group, any hub from an advertisement group may act as the representative of the group. The formation of query groups can be done in a decentralized fashion in several ways. One alternative is again to use some global function that maps hubs to query groups. Another approach is for hubs to pick the members of their query groups based on some metric such as physical distance. A hub may also find its best query group representatives using a "word-of-mouth" approach: it queries other hubs it knows, about representatives they know in the different advertisement groups, and selects the best representative hub based on mutual agreement.

5.1 Advertisement Registration

When an information provider peer wishes to publish a registration on a JXTA search network, it contacts one of the search hubs it knows. If the advertisement group division is not content-based, the advertisement is stored in the local index of the hub. Periodically, each hub sends a summary update to all the hubs in its advertisement group, with all the new additions and deletions of registrations. If the division into groups is content-based, the advertisement is forwarded to the representative of the advertisement group responsible for registering this type of content.

5.2 Query Resolution

Query resolution using the above organization of hubs is performed as follows. When a node issues a query, it sends it to one search hub it knows. The hub searches its local index and the index summary of all the hubs in its advertisement group for a matching advertisement. If the hub finds that another hub in its group may have a match, based on the summary, it forwards the query to that hub. If no match is found in the advertisement group, the query is broadcast to other advertisement groups. If the advertisement groups are organized based on content, the query is forwarded to the representative of the advertisement group responsible for indexing this type of content.

5.3 Advantages of the JXTA Search Architecture Network

The above organization of the hubs can support very dynamic edge peers, advertisement registration updates need to be propagated only to hubs in the same advertisement group. The number of hubs in an advertisement group is a controllable parameter. Moreover, this architecture reduces the query response, since the query needs to be sent to only the representatives of the advertisement groups. Also, because of the centralized architecture of the hubs, it is much easier to implement policies for security, membership and accounting.

6 Conclusion

We reviewed in this chapter some recent approaches for search in distributed networks. We believe that the best distributed search solution depends on the application, the resources and characteristics of the peers in the network, and the presence of business or legal constraints. Given the highly heterogeneous and dynamic environment of a peer-to-peer network, designating a set of "super-peers" which collectively maintain and update an index of all the resources published on the network is generally a good scalable solution.

References

Adamic, L., Lukose, R., Puniyani, A., and Huberman, B. (2001), "Search in Power-Law networks." Available at http://www.parc.xerox.com/istl/groups/iea/papers/plsearch/plsearch.pdf .

Adar, E. and Huberman, B. (2000), "Free riding on Gnutella", Technical report, Xerox PARC, August.

Clarke, I. (2001), "A distributed decentralized information storage and retrieval system." Available online at http://freenetproject.org/freenet.pdf .

Clip2 (2000), "Gnutella: to the bandwidth barrier and beyond," November. Available at http://www.clip2.com/gnutella.html .

Dabek, F., Brunskill, E., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I., and Balakrishnan, H. (2001), "Building peer-to-peer systems with Chord, a distributed lookup service." Available online at http://pdos.lcs.mit.edu/chord .

Decker, K., Williamson, M., and Sycara, K. (1996), "Matchmaking and brokering," *Proceedings of the Second International Conference on Multi-agents Systems (ICMAS-96).*

Druschel, P. and Rowstron, A. (2001), "Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems," *ACM SIGCOM.*

Fan, L., Cao, P., and Almeida, J. (2001), "Summary cache: a scalable wide-area Web cache sharing protocol." Available online at http://www.cs.wisc.edu/~cao/papers/summarycache.ps .

Gibbins, N. and Hall, W. (2001), "Scalability issues for query routing service discovery," *Proceedings of the Second Workshop on Infrastructure for Agents, MAS and Scalable MAS.*

Heimbigner, D. (2001), "Adapting publish/subscribe middleware to achieve Gnutella-like functionality," *ACM Symposium on Applied Computing.*

Houston, B. (2001), "Egocentric self-organization." Available online at http://www.exocortex.org/p2p/egocentric.html .

Joseph, S. (2001), "Adaptive routing in distributed decentralized systems: NeuroGrid, Gnutella and Freenet," *Proceedings of Workshop on Infrastructure for Agents, MAS, and Scalable MAS, at Autonomous Agents 2001*, Montreal, Canada. Available online at http://www.neurogrid.net/php/si-simulation03.zip .

Kleinberg, J. (1999), "The small-world phenomenon: an algorithmic perspective," Cornell Computer Science Technical Report 99-1776.

Peck, M.R. "coderman" (2001), "Decentralized resource discovery in large peer based networks." Available online at http://www .cubicmetercrystal.com/alpine/discovery.html .

Plaxton, C.G., Rajaraman, R., and Richa, A.W. (1997), "Accessing nearby copies of replicated objects in a distributed environment," *Proceedings of ACM SPAA.*

Prinkey, M. (2001), "An efficient scheme for query processing on peer-to-peer networks." Available at http://aeolusres.homestead .com/files/index .html .

Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. (2001), "A scalable content addressable network," *ACM SIGCOM.*

Rohrs, C. (2001), "Query routing for the Gnutella network." Available online at http://www.limewire.com/developer/query_routing /keyword%20routing.htm .

Zhao, B.Y., Kubiatowicz, J., and Joseph, A. (2001), "Tapestry: an infrastructure for fault-tolerant wide-area location and routing," Computer Science Department, University of California, Berkeley Report No. UCB/CSD-01-1141.

This page is intentionally left blank

Chapter 2

Adaptive Content Mapping for Internet Navigation

R.W. Brause and M. Ueberall

This contribution discusses the current state-of-the-art techniques in content-based searching and proposes a particular adaptive solution for intuitive Internet document navigation, which not only enables the user to provide full texts instead of manually selected keywords, but also allows him/her to explore the whole database. Especially, the proposed HADES system supports adaptive classification, i.e., the classification structure is not constant but reflects always the characteristics of the growing document collection. Furthermore, the user interface is augmented by using content based similarity mapping and the possibility of zooming into the content. The client-server implementation stresses the ability for document secrecy and load balancing between clients and servers.

1 Introduction

The Internet as the biggest human library ever assembled keeps on growing. Although all kinds of information carriers (e.g., audio/video/hybrid file formats) are available, text based documents dominate. It is estimated that about 80% of all information worldwide stored electronically exists in (or can be converted into) text form. More and more, all kinds of documents are generated by means of a text processing system and are therefore available electronically. Nowadays, many printed journals are also published online and may even discontinue to appear in print form tomorrow.

This development has many convincing advantages: the documents are both available faster (cf. prepress services) and cheaper, they can be searched more easily, the physical storage only needs a fraction of the space previously necessary and the medium will not age.

For most people, fast and easy access is the most interesting feature of the new age; computer-aided search for specific documents or Web pages becomes the basic tool for information-oriented work. But this tool has problems. The current keyword based search machines available on the Internet are not really appropriate for such a task; either there are (way) too many documents matching the specified keywords are presented or none at all. The problem lies in the fact that it is often very difficult to choose appropriate terms describing the desired topic in the first place.

This contribution discusses the current state-of-the-art techniques in content-based searching (along with common visualization/browsing approaches) and proposes a particular adaptive solution for intuitive Internet document navigation, which not only enables the user to provide full texts instead of manually selected keywords (if available), but also allows him/her to explore the whole database.

2 Standard Information Retrieval Methods

The content based search within text documents has been established under the term *text retrieval*, which historically represents the first and most important branch within the *information retrieval* discipline, and is still subject to intensive research. Although we explicitly focus on *text retrieval* here, please note that almost all underlying concepts reviewed in this chapter can be applied to other information retrieval branches as well: just substitute “terms” and “words” by “features”.

In practice, the most obvious approach to characterize text documents by syntactic and semantic analysis quickly turns out to be intractable at least now. Therefore, almost all of the information retrieval mechanisms are based on condensed representations of the original documents like terms (i.e. keywords or catch-words) or meta information, if available.

2.1 Keyword Search

The most simple approach to a content based search consists in scanning for one or several keywords. Although this is a straight and simple approach, a full text search takes too long for large databases. Therefore, all traditional Internet search engines like *Alta Vista* or *Fireball* parse the visited documents and only search within distilled term lists, which can also be accessed *much* faster (Fellbaum 1998).

This plain keyword search has the disadvantage of hit lists often being either too short or too long, because the user chose either wrong or inadequate keywords or very common terms. Limiting the result set to a reasonable size becomes an art *per se.*

2.2 Recall Enhancers

The first improvement over plain keyword search (known as "aliasing") consists in enlarging the user-provided list of keywords by similar words which can be restricted to cases where the original set of words resulted in too few hits. Common related techniques include:

♦ *word stemming*
Here, all suffixes of words (e.g. in English (Porter 1980) or German (Caumanns 1998)) are discarded. Errors occur, if the same word stem is obtained for words of different semantics (*overstemming*) or if different stems are obtained for the same semantics (*understemming*). Note that all stemmers (unless they

are dictionary-based) are language-dependent: an inappropriate use, e.g. in mixed language documents, leads to a drastic reduction in stemming quality.

- *dictionary-based identification of synonyms*
  A thesaurus is very useful if you want to cope with the problem of existing words having the same meaning (*synonymy*) or the same word having different meanings (*polysemy*), e.g. the word *bank* which has about a dozen different meanings in English.
- *synonym sets*
  Another idea consists in the construction of a set of all nouns, verbs, adjectives and adverbs, associated with a single semantic meaning (which can be seen as some sort of super-thesaurus). A freely available realization of this concept is the WordNet lexical reference system (Fellbaum 1998). It requires a much higher degree of manual input than a thesaurus and therefore cannot be generated semi-automatically as it is possible with the latter.

Additionally, all keywords (either user-provided or derived by aliasing) can be weighted or modified by means of a concept known as *relevance feedback* (Harman 1992, Leuski 2000b), whereby the user iteratively rates some documents in the list of results as being relevant or non-relevant. The system then tries to modify the keyword list accordingly, e.g. by discarding those query terms which only occur within non-relevant documents.

2.3 Using Meta Information

A lot of meta information is contained in texts which contain semantic markup information, e.g. HTML/XML based Web pages with title or paragraph headers in subsequent order. The HTML standard also defines a "`meta`" tag which can (and should) be used to provide special information, e.g. the author's name, manually identified keywords, or even an entire abstract.

Of course, these entries are of use only when they obey to a common standard (e.g. the Dublin Core Metadata Standard (1995)), and cannot easily be maintained for an exponentially increasing number of documents which nowadays are often – at least partially – automatically generated by querying Web-based databases. Here, automatically derived meta information is needed.

One of the traditionally used sources of meta information are citations which can easily be used to build an automated ranking system, see e.g. *CiteSeer* (Giles *et al.* 1998). Another feature of XML/HTML based documents, which distinguishes them from those in plain text format, is the existence of (true bi-directional) hyperlinks which can be taken into account for this task, too. Examples are the PageRank measure introduced by Google (Brin and Page 1998) or the more flexible HITS concept (Kleinberg 1998).

As of today, an automatic ranking of documents *solely* based on user-provided meta data greatly suffers from the overall imprecision and sponginess of the latter.

2.4 Classification Trees

One important alternative for searching an unstructured set of documents is the manual classification of the document and the arrangement of the classes in a tree-like manner. These *classification trees* are very common in traditional library work and provide some advantages. They facilitate the search for new, unknown sources just by browsing through an appropriate subcategory of the classification tree. Even if the unknown document does not contain the critical keywords or terms, it can easily be found provided that the right paths to topics of interest are identifiable by the user.

The disadvantages are also well known:

♦ Since the classification is fixed, it is difficult to introduce changes in the classification system for evolving subjects, e.g.

technology. It might become necessary to reorder whole subtrees which requires a reclassification of all the documents in the subtree.

- Documents often cannot be assigned to one single category. The common remedy for this problem is the duplication of the document reference which produces a multi-class membership. This implies other problems, see (Kaszkiel and Zobel 2001).
- The exploring and browsing is impeded by the fixed subclass boundaries and cannot automatically be redirected across the branches of the classification tree to another relevant branch.
- If many documents are located below one single subcategory, there is no feasibility to discriminate between them; in this case, the corresponding node can only be referred as a whole.
- The classification has to be performed manually by humans which is not affordable for huge collections. This is the crucial problem of Internet based documents: For the exponentially growing Internet resources, manual processing is prohibitive. Although there are initiatives like the *Open Directory Project* (http://dmoz.org) which involves thousands of librarians, an automatic classification is necessary. Today, most of the automatic classification efforts rely on the automatic extraction of document features. This is done by a process called *indexing*.

2.5 Indexing

Standard information retrieval approaches use keywords, stemmers and thesauri only as preprocessing filters. They try to represent a document solely by all distinct terms which might characterize it. For this purpose, you have to preprocess and condense a document by several steps (Blelloch 1998):

(1) choice of appropriate terms

From the document choose appropriate terms and include them in a term collection. The meaning of "appropriate" depends on the chosen concept. For instance, the most simple strategy is to drop frequently occurring, uninteresting words (*stop terms*) like "and", "to", "in" (Sahami 1998). This commonly used technique has the disadvantage that combinations of these simple words (phrases), e.g. "to be or not to be", might be important, and could not be found if the isolated words were discarded as stop terms. There are frequently used words which also should not be dropped like "time", "war", "home" (Frakes 1992). Certainly, this makes the selection more subjective, or, at least, domain-dependent.

A more evolved strategy (Salton *et al.* 1976, 1981) only selects those terms that have a high *discrimination value* within a given set of documents. Here, as objective function to be minimized, the average document similarity based on the chosen subset of terms is used. Unfortunately, the computation of all interactions (similarities) between all documents is computationally expensive. This can be reduced by replacing the average similarity between all documents by the average similarity between the documents and the term prototype, the average weight of all document terms. For each step of selecting the most discriminative terms, the value of the objective function is computed before and after dropping (or including) a term. If the objective function is increased by dropping a term, its discrimination value is positive and the term should be retained. Otherwise, it can be dropped.

Alternatively, instead of trying to identify (good) index *words* in the first place, simple *substrings* of fixed size *N* (so called *N-grams* (Damashek 1995)) can be extracted from a given text. If hash tables are used to store the counters needed to calculate the relative frequency of these substrings within the document after parsing, this approach is very fast (Cohen 1997a). Aside from this, the concept of N-grams is clearly language-independent, but, on the other hand, also not very descriptive for humans.

(2) weighting of the terms

Long documents naturally contain the same terms more frequently than short ones. In order to get rid of this peculiarity, the term frequencies have to be normalized (Wulfekuhler and Punch 1997). Also, long documents contain a higher number of distinct terms which might better match a given request. Therefore, the length of a document has to be taken into account when weighting the terms.

(3) choice of appropriate indexing data structures

There are two popular data structures for index management: *signature files* and *inverted index files.* For each document, a signature file is created which consists of hash-encoded bit patterns of text parts within the corresponding document. This drastically reduces the search time, because instead of the document itself, only the much shorter signature file is searched for the hash-encoded search terms (Faloutsos 1992).

Alternatively, we might invert our term lists, one for each document, by building global lists, one for each term of the "global vocabulary", i.e. all distinct terms within the collection. Each list (*posting list*) contains the pointers to documents that include the specific term. This method has a lot of advantages over the use of signature files (e.g. false matches cannot occur) and should be preferred, see (Zobel and Moffat 1998).

How many different terms do we have occurring f_l times? To evaluate this, let us order all terms according to their occurrence frequency and assign them an index. The index one is for the most frequent term, the index two to the next frequent one and so on. Then we will notice an interesting fact: the product of index r and frequency f is approximately constant: $r{\cdot}f$ =const=K. This observation is known as "Zipf's law" (Zipf 1949). The number of different terms with frequency f_l is reflected by the number of indices which have the same number f_l of terms, the difference $\Delta r = r_2 - r_1$

$=K(\frac{1}{f_1+1}-\frac{1}{f_1}) = K\frac{1}{f_1(f_1+1)}$. The constant K can be observed for the rank r_m with frequency $f=1$: $K= r_m \cdot f = r_m$.

In conclusion, Zipf's law says that the number of different terms increases nonlinearly with decreasing term occurrence and importance. Therefore, an important fraction of terms can be dropped if we introduce an occurrence threshold for the list of terms.

2.6 Vector Space Models

One of the classical methods of encoding the information space of a given set of documents is the approach of applying the well-known mathematical tool of linear algebra. Regarding the entries d_{ij} as the components of a document vector relative to "term vectors", the vector space model (VSM) (Salton *et al.* 1975) describes the documents as a linear combination of orthogonal base vectors, representing the basic terms.

Given a static vocabulary consisting of n distinct terms, each document can be represented as a vector of length n. Therefore, the documents as rows of terms form a document-term matrix $\mathbf{D}$ with the terms as columns. Each entry d_{ij} in the matrix represents the number of occurrences of term j within document i.

The "formally unclean" assumption of orthogonal base vectors was remedied by the later-proposed generalized vector space model (GVSM) (Wong *et al.* 1985). Here, the orthogonality of boolean minimal conjunctive expressions (the dual space) is exploited to generate orthogonal base vectors.

Later on, the GVSM approach was modified to only represent the most relevant linear combinations of document features by Latent Semantic Indexing (LSI) (Deerwester *et al.* 1990) and to drop "unimportant" correlations. The term correlations between documents are treated by their statistical properties: the document-term matrix

D is analyzed by a singular value decomposition in order to reduce the number of descriptive dimensions and to get the principal directions as intrinsic latent semantic structures.

2.7 Similarity Measures

Within the mentioned vector space models, a query can be regarded as the problem of finding the most similar document to a given *pseudo-document* (e.g. consisting of a user-provided list of keywords or a *real* document). The similarity measures employed here are often derived from standard linear algebra measures, for instance the scalar product or the cosine between the vector representations of the documents to be compared (Salton and Buckley 1988, Zobel and Moffat 1998).

Here, for our purpose a less-known (but not less-compelling) measure, the *cover coefficient concept* (CCC) (Can and Ozkarahan 1983, 1984, 1985), shall be sketched. Defining the importance of the j-th term relative to all terms of document $\mathbf{d}_i$, the *i*-th row of **D**, by

$$s_{ij} = \frac{d_{ij}}{\sum_k d_{ik}} \tag{1}$$

and the importance of the *j*-th term in document $\mathbf{d}_i$ relative to all documents in the collection containing the term (*j*-th column of **D**) by

$$\tilde{s}_{ij} = \frac{d_{ij}}{\sum_k d_{kj}} \tag{2}$$

we get the degree c_{ij} of document coverage (the cover coefficient matrix **C**) from the cross-correlation between two documents

$$c_{ij} = \sum_k s_{ik} \tilde{s}_{jk} \tag{3}$$

One major problem of all similarity measures discussed in this section is the situation where new documents with unknown terms have to be inserted in and compared with an existing collection of documents. Here, for huge collections the length of the vectors (list of terms) usually become very long and both the comparison and the weighting process becomes intractable. One approach to deal with this problem consists in passing over from a global document description into a local one which is only valid within a certain context or cluster of documents discussed in the next section.

3 Adaptive Content Mapping

The most interesting alternative to the manual classification task is the automatic, content based classification which maps the documents into different classes. In general, the topic oriented associative relationship maps have been standardized by the international norm ISO 13250, see e.g. (Topic Maps 2001), but there is no standard adaptive approach. Although the actual adaptive methods still have problems, the rapid growing Internet content produced by non-librarians allows no other approach in the near future. Based on the methods introduced in the previous sections, we will briefly review current adaptive content mapping methods.

3.1 Automatic Classification by Clustering

The similarity measures defined so far can be used to group documents into clusters. These semantic clusters represent a natural classification. In contrast to the static classification performed according to fixed criteria in Section 2.4, the adaptive classification reflects the statistical properties of the document collection and will change according to the specific document collection.

There are two kinds of clustering algorithms: the non-hierarchical ones which, given a neighborhood criterion and a distance metric, divide the document space into a set of clusters, and the hierarchical ones which find clusters composed of smaller clusters on several levels. For an overview, see (Rasmussen 1992, Willett 1988).

Here, we take a closer look at the non-hierarchical cluster algorithms using the cover coefficient concept. With the expected number n_c of clusters

$$n_c = \left\lceil \sum_i c_{ii} \right\rceil \tag{4}$$

and the "cluster seed power" measure, which basically tries to capture the extend with which terms are distributed within a set of documents (Can and Ozkarahan 1989) and can be used to derive the term discrimination value of individual terms as well as to identify documents which contain a high number of "good" terms. The algorithm can be summarized as follows:

```
1. $N_c$:=0; WHILE $N_c<n_c$ DO
      Choose ($n_c$–$N_c$) the next documents of maximum cluster seed
         power as new cluster seeds
      Let $N_c$ be the number of equivalence classes within this
         (sub)set of documents (two documents i and j belong to the
         same class if they have nearly identical $c_{ii}$, $c_{iij}$ $c_{ji}$ and $c_{ii}$)
   ENDWHILE
2. With the $N_c$ cluster documents obtained, assign each document i
   of the collection not being a cluster seed to the cluster document
   k of maximal coverage $c_{ik}$.
3. Documents which were not assigned to any cluster during the
   last step form a cluster by themselves.
```

This cluster algorithm has several advantages:

- It is stable; small variations in the term-document representation only lead to small changes in clustering
- If there is no similarity between documents, they will not share the same cluster as opposed to standard algorithms

- Given *m* documents and *n* terms, *n*>>*m*, this algorithm will cluster the documents by a computation complexity of $O(m \cdot n)$
- The input sequence of the documents does not influence the clustering results

3.2 Adaptive Hierarchical Classification

The non-hierarchical cluster methods produce a set of clusters without any structure. For huge sets, the navigation is greatly facilitated if the set can be structured in a hierarchical manner. An automatic hierarchical classification, adapted to a document collection, can be performed by two different approaches, either bottom up or top down:

- *Agglomerative approach*
  The agglomerative strategy tries to fuse small entities in order to get bigger ones on the next higher level. The clustering fuses the *m* documents by *m-1* operations into a tree structured cluster set. A common used algorithm for this is the *nearest neighbor* approach (Rasmussen 1992, Willett 1988).

- *Divisive approach*
  The division of each cluster into smaller clusters is based on the similarity measure between the documents.

  One of the algorithms for successively dividing clusters and grouping them in a tree is the Principal Direction Divisive Partitioning algorithm (PDDP) introduced by Boley (1997). Like the LSI algorithm of Section 2.6, it uses the dominant eigenvectors of the appropriate cross-correlation matrix. It transforms the document descriptions of the most scattered cluster in the *eigenspace*, and, based on the principal eigenvector, then decides for each document of the cluster whether to shift it in either the left or the right leafs of a binary tree.

> The algorithm was developed in the context of the WebACE project (Boley *et al.* 1999), where an user agent automatically retrieved potentially relevant documents from the web, based on a single user profile (namely, bookmarks and visited pages).

One of the newer search engines using hierarchical clustering is the Vivísimo project (http://www.vivisimo.com). It uses conventional search engines for keyword search and then clusters the results dynamically, notably without parsing the referenced documents in its entirety by itself.

3.3 Local Adaptation

Once the document collection has been transferred into a hierarchical classification, it becomes very expensive to add new documents. In the extreme, by statistical deviations, the whole adaptive classification tree becomes unstable and has to be reorganized. How do we handle such a situation?

In principle, this cannot be avoided if we want the classification to properly reflect the data-induced configuration. Nevertheless, we can try to make unstability less probable by several means:

- During initial adaptive clustering and classification, the documents with the "broadest" set of features should be chosen in order to build up a very general framework.
- For subsequent insertions of new documents, the structure should be kept stable as it is in the case of manual classification of Section 2.4.
- The whole process of adaptive classification might be resumed if the number of new documents exceeds a predefined threshold.

This approach has one major drawback: the high computational costs of reorganizing the whole document collection, even if it occurs only periodically. As a compromise between stability and plasticity, only local adaptations can be made. This kind of continuous re-adaptation avoids the complexity of adapting the whole collection and supports the correct local document relations. Nevertheless, in the case of huge local changes in document statistics also changes in the global class hierarchy have to be considered.

4 Intuitive Navigation

User interfaces for smart ("intelligent") systems have to face many demands. One of the most popular is described by the term "intuitive" which is not well defined. Raskin (2000) references it as "familiar" which means that the guessing in bad user interfaces is replaced by knowledge. In this sense, we want to implement a user interface which is based on already existing knowledge.

4.1 Hierarchical Navigation and the Zoomable User Interface

The search in huge databases is often facilitated by the approach of successively splitting the search space into smaller parts. This divide-and-conquer approach only needs logarithmic time, in contrast to an exhaustive scan of the entire database. It can be backed up and exploited by the user interface design. For browsing through a huge database, you might structure the data in a hierarchical manner and use the hierarchy in the user interface. Each hierarchy level might be presented visually, in a way appropriate to its content. On each hierarchy level, the user decides where to go next and selects the next level until he/she reaches the underlying document(s).

This idea of a level oriented top-down (and vice versa also bottom up) user interface can be extended to a continuous version: the zoomable interface (Bederson and Hollan 1994, Bederson and Meyer 1998, Raskin 2000). This interface propagates the idea that the metaphor of flying, approaching a place by zooming in and leaving a place by zooming out, is sufficient to navigate within huge databases.

The zooming interface only has two modes: shifting and zooming. When you shift within an hierarchical level you only see abstract quantities mapped into a 2D plane. You move within these entities and select an interesting region. Then, you switch to zooming and approach the spot (a document) while the context (the other documents) becomes clearer, and you might even deviate to a more appropriate document.

The zoomable interface can also be used for other purposes than database navigation. Raskin (2000) claims that it is even capable of replacing the traditional user interface completely, thereby rendering mouse devices and windows superfluous.

4.2 Similarity Based Visualization

There are already systems for intuitive navigation in documents by means of graphical user interfaces. One of the most straightforward implementations of content based navigation consists of placing all documents as symbols (small rectangles or circles) on a 2-D plane. The location on the plane is chosen according to their similarity value based on index terms, see Section 2.7. There are several approaches for determining the position of a document (or document cluster) within the plane.

The first approach is given by the vector space model: each document is described by an index term vector of length n which gives the absolute coordinates or, alternatively, the difference, i.e., the relative position between the documents to set up the 2-D display. This approach also needs a mapping stage where the n-dim. document space is mapped on the 2-D display.

One of the classical algorithms for doing this mapping is the nonmetric algorithm for multidimensional scaling (MDS) (Kruskal 1964a,b), which is computational expensive. A fast heuristic can be found in (Faloutsos and Lin 1995). Here, the objective function "stress" (this term really represents a family of functions, cf. Cohen

(1997a)), a measure of difference between the original distance matrix and the 2-D distance matrix, is minimized. One of the most compelling definitions for stress within this family is the so-called "proportional stress" which punishes deviations at long distances proportionally more than those at small distances. This can be interpreted as proportional to the energy of a system of particles joined by springs whose equilibrium configuration corresponds to a local energy minimum. Therefore, a system of "force-directed placements" like this used for visualizing a graph is often called a *spring embedder* and very popular in graph visualization. As application example, the *Lighthouse* system display of 50 documents matching the query "Samuel Adams" is shown in Figure 1. The 50 matching documents are visualized as pseudo 3-D balls, the best matching ones marked by thick circles. Since the original text references are included, the whole window quickly becomes overloaded. Since these algorithms do not consider absolute coordinates, the resulting picture has no preferred orientation; it can arbitrarily be rotated.

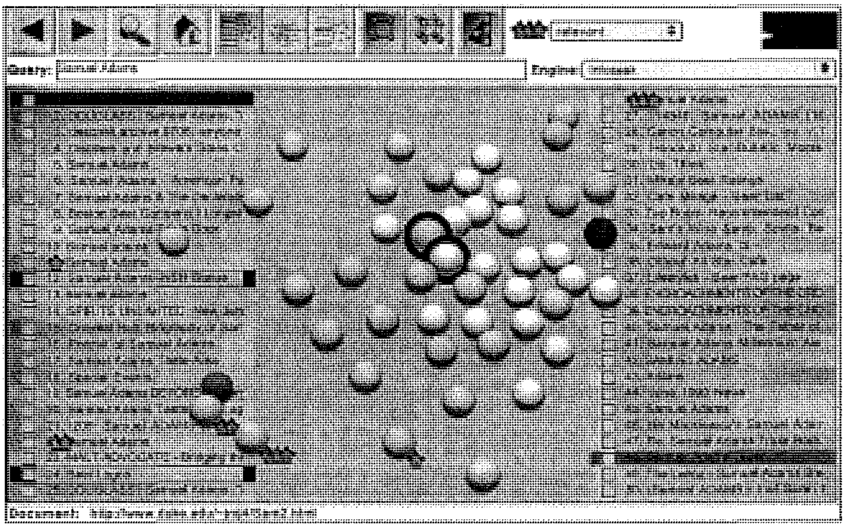


Figure 1. A query display of the *Lighthouse* system (Leuski 2000a).

This interface suffers from the several drawbacks:

- Only documents are displayed which match a certain search criterion, all other documents are ignored.
- The configuration of displayed documents change after each modification of the search criterion, making it impossible to remember a certain area of the document space.
- The number of documents in the display is limited to approximately 100.

Therefore, huge document collections can hardly be explored. As a remedy, hierarchical maps may be defined. One of the most famous examples is the WEBSOM approach (Honkela *et al.* 1996) where an adaptive Kohonen map is used for mapping the document space onto a regular 2-D grid. The contents of the nodes in the fixed display has to be evaluated afterwards. In Figure 2, a couple of windows representing several hierarchical levels are shown.

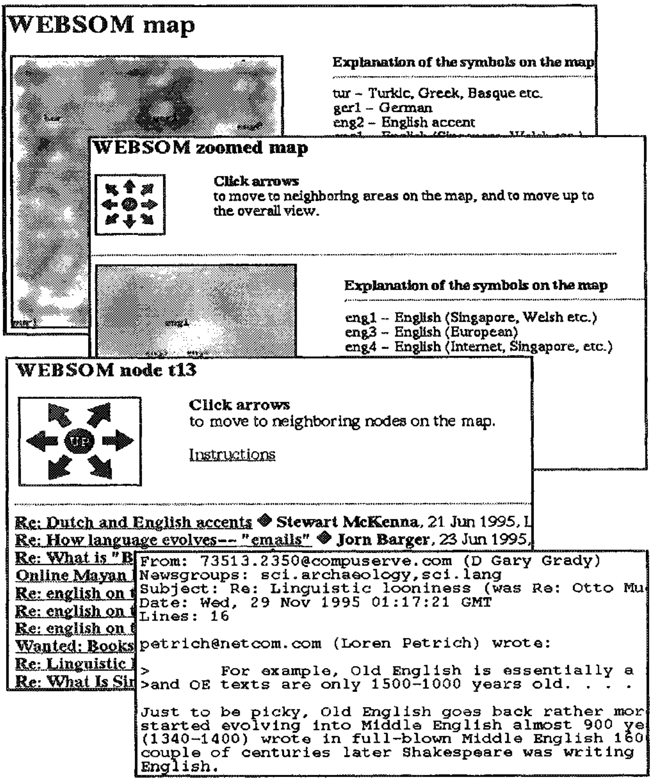
This absolute coordinate approach has several disadvantages:

- If you introduce new documents and/or new terms, the whole system has to be retrained which takes a long time in huge databases – often prohibitively long.
- Another disadvantage is that the layout will change afterwards. Since the cluster display changes, the user has to habituate to the new scene even when he/she already knows the majority of documents.

Here, too, some properties hinder an intuitive navigation:

- The high number of windows of several search process "levels" makes it difficult to maintain an overview of the search process
- The *document content distance* between the regular spaced clusters in the map display are expressed by different color shades. However, this makes a quick orientation rather difficult.

An interesting alternative visualization is demonstrated by the *WebMap* system (http://www.webmap.com), which allows for the (manual) assignment of icons to clusters and single documents.



WEBSOM map

Explanation of the symbols on the map

tur – Turkic, Greek, Basque etc.
ger1 – German
eng2 – English accent

WEBSOM zoomed map

Click arrows
to move to neighboring areas on the map, and to move up to the overall view.

Explanation of the symbols on the map

eng1 – English (Singapore, Welsh etc.)
eng3 – English (European)
eng4 – English (Internet, Singapore, etc.)

WEBSOM node t13

Click arrows
to move to neighboring nodes on the map.

Instructions

Re: Dutch and English accents ◆ **Stewart McKenna**, 21 Jun 1995, L
Re: How language evolves-- "emails" ◆ **Jorn Barger**, 23 Jun 1995,
Re: What is "B
Online Mayan
Re: english on t
Re: english on t
Re: english on t
Wanted: Books
Re: Linguistic
Re: What Is Si

```
From: 73513.2350@compuserve.com (D Gary Grady)
Newsgroups: sci.archaeology,sci.lang
Subject: Re: Linguistic looniness (was Re: Otto Mu
Date: Wed, 29 Nov 1995 01:17:21 GMT
Lines: 16

petrich@netcom.com (Loren Petrich) wrote:

>       For example, Old English is essentially a
>and OE texts are only 1500-1000 years old. . . .

Just to be picky, Old English goes back rather mor
started evolving into Middle English almost 900 ye
(1340-1400) wrote in full-blown Middle English 160
couple of centuries later Shakespeare was writing
English.
```

Figure 2. The WEBSOM adaptive map and its hierarchical windows (Honkela *et al.* 1996).

5 The HADES System

In this section we will present a new adaptive system for intuitive navigation called HADES (Hierarchical Adaptive Document Exploration System). Its underlying concepts are based on the review results presented in the previous sections, integrating the most advanced and our new concepts into one concise design and adding often neglected but important features like portability and intelligent load balancing (Ueberall 2001).

5.1 Specifications

The system has to meet the following criteria:

- *Adaptive classification*
  The classification structure must not be constant but always should reflect the characteristics of the growing document collection.

- *Intuitive navigation*
  The user interface should reflect the underlying hierarchical structure. It should be possible to explore the classification tree intuitively (i.e. without special training).

- *Modularity*
  The system has to be designed such that it contains functionally distinctive modules which enables local updates of functions or even the complete replacement of them by similar functional software.

- *Portability*
  The program code should not depend on a specific machine type or operating system but should be easily portable to new architectures.

- *Load balancing*
  For large document collections, the interaction speed and therefore the user acceptance of the system depends on the ability to automatically distribute the workload within a cluster of servers. This feature can hardly be implemented afterwards – it has to be taken into account at specification time.

5.2 The Adaptation Mechanisms

There are several mechanisms which are designed to reflect the specification of adaptive classification:

- *Adaptive clustering*

  When a new document is processed by the system, at first all terms are extracted by the parser. This reduced representation is then merged into a central data structure, consisting of a number of inverse index tables (Blelloch 1998, Moffat and Zobel 1996, Zobel and Moffat 1998), see Section 2.5. This enables the inclusion of new distinct terms of new documents. Then, the document is routed, starting with the root node of the classification tree, until the lowest hierarchy level (leaf) is reached and inserted in the last node visited. The similarity measure for routing is the so-called cosine coefficient (Moffat and Zobel 1996, Salton *et al.* 1975), combined with the cover coefficient concept (Can and Ozkarahan 1985).

  If the node cluster size limit is reached, the cluster must be split into several parts. During this operation, the involved node has to be locked and the reorganization takes place.

  The more interesting case, which involves the fusing of nodes, is computationally much more expensive: If the node contains references to other nodes, instead of fusing the whole collection and completely reorganizing it, we use the following heuristic: We do not lock all document representations but use *copies* of a *subset* of them when re-clustering. Afterwards, the resulting clusters are split. If the new clusters contain too many representatives of different clusters, a shifting and reorganization is not favorable. Instead, the same algorithm is recursively tried on lower levels until it either reaches a smaller diversity in a cluster or the lowest node of the tree. This kind of heuristic assists the demand for structure conservation and *recognition support* for the user (Frakes 1992).

  Note that the original representatives (and subtrees) of the hierarchy are not touched until the re-clustering was successful in which case a quick node substitution takes place. Otherwise, large parts of the hierarchy would permanently be inaccessible to the users while updating the database.

The dynamic, adaptive hierarchy depends on the sequence ordering of the incoming documents. This might result in the paradox that the same document is assigned to two different leafs of the classification tree, depending on the time when it has been classified. Reclassifying the whole collection after classification changes may be prohibitive for large collections and will necessitate unwanted reorientation efforts of the user.

Here, a compromise between stability and plasticity has to be designed. The kind and degree of adaptation has to reflect the users' needs for stable, known classification regions. This is done by the introduction of a *cohesion* and an *adhesion* parameter which depend on the position and depth of the nodes within the hierarchy tree. An additional *affinity* parameter controls the local readapting in regular intervals depending on the workload. All three parameters are controlled by the user habits and adapt to the users' needs.

- *Recognition of structures*
  The goal of intuitive navigation in the context of adaptive clustering demands stable document cluster structures which can be recognized by the user. This avoids confusion at the user interface level and supports the feeling of familiarity with the system.

  Since we have different words which have the same meaning a thesaurus can help to cluster similar documents with different terms into a content based neighborhood. The small document specific thesaurus is automatically generated on the base of a general thesaurus and is treated like an abstract of the document.

- *Meta information*
  Meta information (references of all kind) is preserved during the indexing process and flows into the affinity values for document pairs and clusters. There are still some questions open: What should we do with documents which are referenced by other documents? Should the information be propagated to other lev-

els and if so, how should it be considered there? Should we allow the user to jump back-and-forth between hierarchy nodes?

5.3 Intuitive Navigation

For the exploration of the document database we chose the content based, zoomable user interface as interaction paradigm. It consists of the following elements:

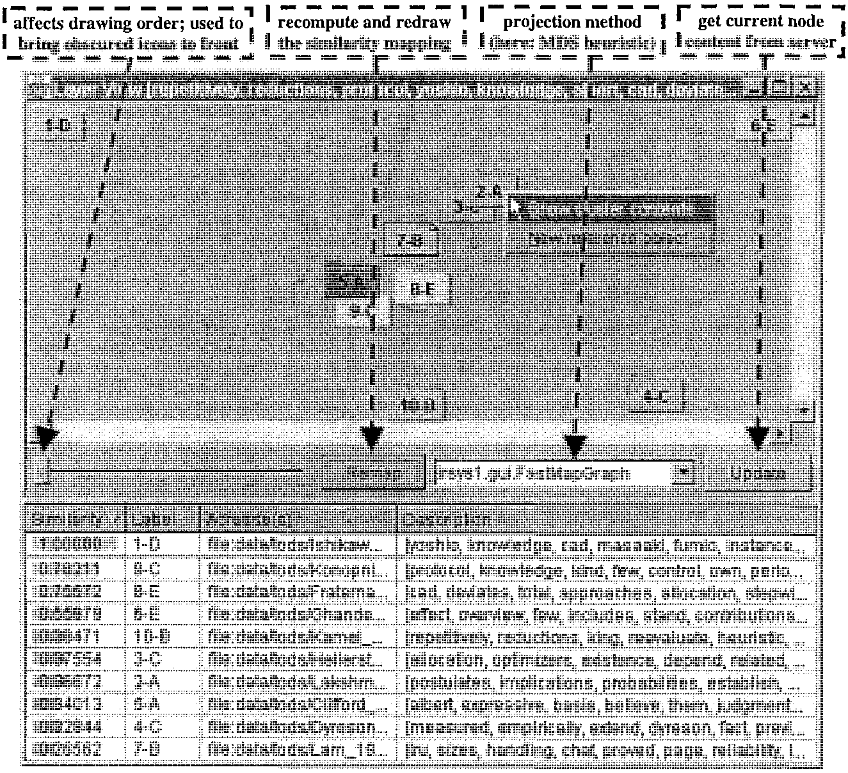
- *content based similarity mapping*
  The documents are represented on a 2D screen window by symbols: sheets for real documents and directory symbols for cluster representatives. The display of the symbols uses the computational feasible FastMap (Faloutsos 1992) algorithm. The distances between the symbols reflect the similarity in document content. As similarity measure we use the well-known *cosine coefficient* (Can and Ozkarahan 1985) in (indirect) combination with the *cover coefficient concept* cf. Section 2.7).

  Additionally, in the extended view the document file information is also shown in a list, ordered by the search request similarity criterion. Figure 3 shows a sample window.

- *zoomable content*
  The zoomable interface permits the display of details if you zoom into a document. In order to implement this, we chose not to present the document text directly in physically different resolutions to the user, but to successively show the document details in several stages: In the first stage only a main term, then a term list, then an abstract and afterwards the whole text are shown. The abstracts or relevant text fragments (*gists*) are generated automatically, see (Cohen 1995, Liddy 1994).

- *zoomable hierarchy*
  For the representative documents, we have to distinguish between the document itself and its representation function. In the latter case we switch to the next level of hierarchy and its asso-

ciated similarity mapping. In Figure 4 this is shown for the example of Figure 3 where the cluster representative "2-A" has been selected. Please note that the node description has adaptively changed due to the new context showing the new discrimination terms.

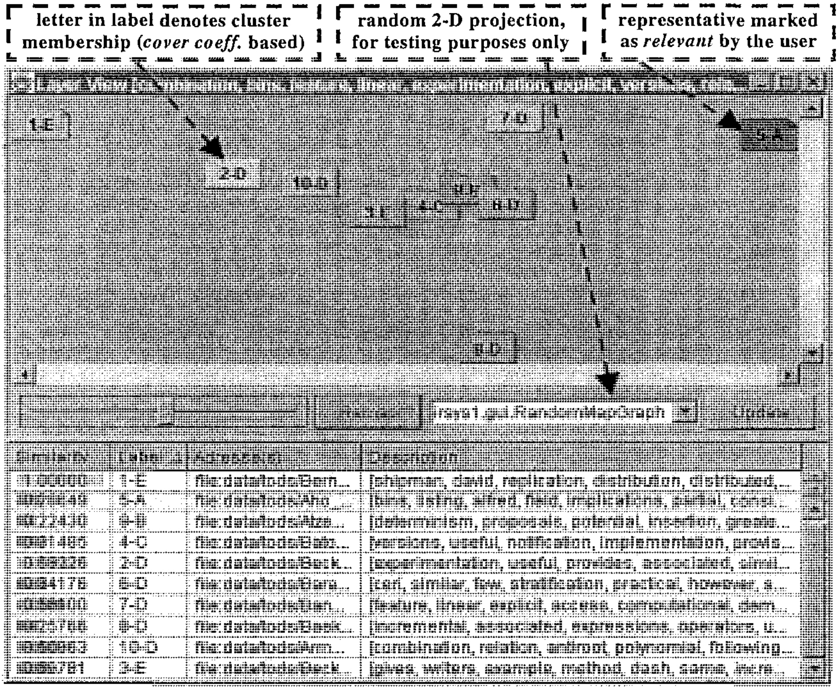

affects drawing order; used to bring obscured items to front
recompute and redraw the similarity mapping
projection method (here: MDS heuristic)
get current node content from server
1-D
6-E
2-A
3-C
New reference point
7-B
5-A
8-E
9-C
10-B
4-C
Redraw
fsys1.gui.FastMapGraph
Update

Similarity	Label	Address(es)	Description
1.00000	1-D	file:data/todo/Ishikaw...	[roshio, knowledge, cad, masaaki, tomio, instance...
0.78211	9-C	file:data/todo/Koeppl...	[protocol, knowledge, kind, few, control, own, perio...
0.75572	8-E	file:data/todo/Fraterna...	[cad, deviates, total, approaches, allocation, stepwi...
0.65979	6-E	file:data/todo/Shande...	[effect, overview, few, includes, stand, contributions...
0.60471	10-B	file:data/todo/Kamel_...	[repetitively, reductions, king, reevaluate, heuristic, ...
0.57554	3-C	file:data/todo/Hellerst...	[allocation, optimizers, existence, depend, related, ...
0.56572	2-A	file:data/todo/Lakshm...	[postulates, implications, probabilities, establish, ...
0.54013	5-A	file:data/todo/Clifford_...	[albert, expressive, basis, believe, them, judgment...
0.52844	4-C	file:data/todo/Dyreson...	[measured, empirically, extend, dyreson, fast, pred...
0.50562	7-B	file:data/todo/Lam_18...	[nu, sizes, handling, chat, proved, page, reliability, l...

Figure 3. Visualization of single documents (sheets) and clusters (directories).

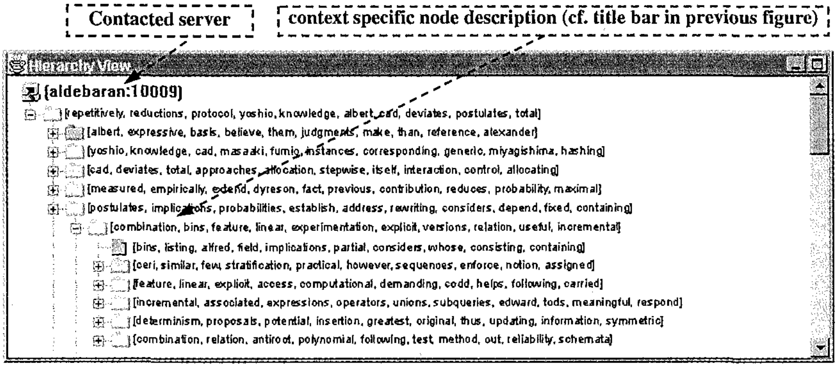
- *context display*

  For navigation, it is often very helpful to orient oneself along the context map in order to plan the next moves. Here, we chose the classification tree as context. A sample display is shown in Figure 5.


letter in label denotes cluster membership (*cover coeff.* based)
random 2-D projection, for testing purposes only
representative marked as *relevant* by the user
1-E
2-D
10-D
7-D
3-E
4-C
8-D
5-A
rsys1.gui.RandomMapGraph

Similarity	Label	Address(es)	Description
1.00000	1-E	file:data/tods/Bern...	[shipman, david, replication, distribution, distributed,...
[illegible]	5-A	file:data/tods/Aho_...	[bins, listing, alfred, field, implications, partial, consi...
[illegible]	9-B	file:data/tods/Alze...	[determinism, proposals, potential, insertion, greate...
[illegible]	4-C	file:data/tods/Babi...	[versions, useful, notification, implementation, provis...
[illegible]	2-D	file:data/tods/Beck...	[experimentation, useful, provides, associated, simil...
[illegible]	6-D	file:data/tods/Bera...	[ceri, similar, few, stratification, practical, however, a...
[illegible]	7-D	file:data/tods/Ben...	[feature, linear, explicit, access, computational, dem...
[illegible]	8-D	file:data/tods/Beck...	[incremental, associated, expressions, operators, u...
[illegible]	10-D	file:data/tods/Arm...	[combination, relation, antiroot, polynomial, following...
[illegible]	3-E	file:data/tods/Beck...	[gives, writers, example, method, dash, same, incre...

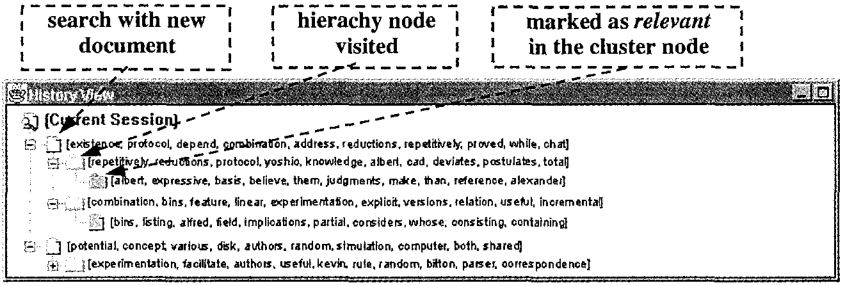
Figure 4. The next hierarchy level.


Contacted server
context specific node description (cf. title bar in previous figure)
Hierarchy View

{aldebaran:10009}
- [repetitively, reductions, protocol, yoshio, knowledge, albert, cad, deviates, postulates, total]
  - [albert, expressive, basis, believe, them, judgments, make, than, reference, alexander]
  - [yoshio, knowledge, cad, masaaki, fumio, instances, corresponding, generic, miyagishima, hashing]
  - [cad, deviates, total, approaches, allocation, stepwise, itself, interaction, control, allocating]
  - [measured, empirically, extend, dyreson, fact, previous, contribution, reduces, probability, maximal]
  - [postulates, implications, probabilities, establish, address, rewriting, considers, depend, fixed, containing]
    - [combination, bins, feature, linear, experimentation, explicit, versions, relation, useful, incremental]
      - [bins, listing, alfred, field, implications, partial, considers, whose, consisting, containing]
      - [ceri, similar, few, stratification, practical, however, sequences, enforce, notion, assigned]
      - [feature, linear, exploit, access, computational, demanding, codd, helps, following, carried]
      - [incremental, associated, expressions, operators, unions, subqueries, edward, tods, meaningful, respond]
      - [determinism, proposals, potential, insertion, greatest, original, thus, updating, information, symmetric]
      - [combination, relation, antiroot, polynomial, following, test, method, out, reliability, schemata]

Figure 5. The hierarchy display window for the example.

- *search history display*
  An additional help is the display of the search history. It shows all documents marked as relevant in a compact form, see Figure 6. Only three levels of hierarchy are displayed: The initial search document, all visited (visualized) nodes and all documents marked as "relevant" within these nodes.



```
[Current Session]
[existence, protocol, depend, combination, address, reductions, repetitively, proved, while, chat]
    [repetitively, reductions, protocol, yoshio, knowledge, albert, cad, deviates, postulates, total]
        [albert, expressive, basis, believe, them, judgments, make, than, reference, alexander]
    [combination, bins, feature, linear, experimentation, explicit, versions, relation, useful, incremental]
        [bins, listing, alfred, field, implications, partial, considers, whose, consisting, containing]
[potential, concept, various, disk, authors, random, simulation, computer, both, shared]
    [experimentation, facilitate, authors, useful, kevin, rule, random, bitton, parser, correspondence]
```

Figure 6. The search history display window.

Additional navigation possibilities evolve if hyperlinks can be exploited to jump back-and-forth between documents. It is not clear if this feature is helpful or confusing and has therefore to be evaluated.

5.4 Implementation Issues

There are several practical implementation features of our system which should be mentioned here also.

5.4.1 Modularity

The code of the system is based on a client-server structure, each one divided into several, independent parts, see Figure 7. Each part can be replaced by an equivalent functional entity implementing another algorithm. Therefore, changes in the database or user interface are easily tolerated.

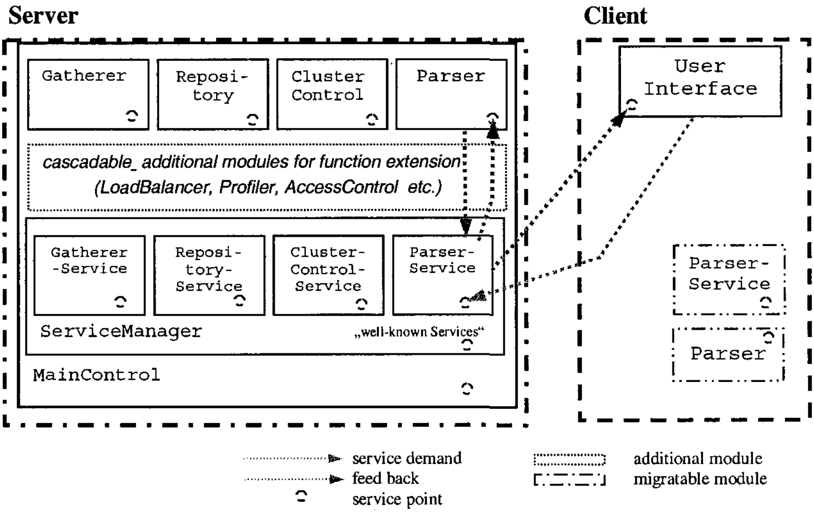


Figure 7. Schematic overview of the system architecture.

The communication between the following listed modules is based entirely on message passing.

- `UserInterface`: This module implements the graphic user interface on the client side.
- `MainControl`: The main module initializes all services on the server side. The server may be part of a cluster.
- `ClusterControl`: This module is responsible for the generation and adaptive modification of the hierarchy of the local computer.
- `Parser`: The parser (on server side) scans documents, transforms them into the internal representation and generates the index terms. This might also be migrated to client side.
- `Gatherer`: This module is responsible for the setup of the internal data structures and preprocesses all document input (scanning for further references).
- `Repository`: this module is responsible for the persistent storage of the internal data structures (documents, meta data and so on.).

The advantages of such a modular concept are obvious:

- The *user interface* can be coupled with a diversity of search engines. It provides an uniform interface based on a 2-D similarity display if possible, or a (conventional) list representation otherwise.
- The replacement of the cluster component of the `ClusterControl`-module allows the use of statistical classifiers.
- The parser can be extended for other data formats independently of the rest of the system.
- The encapsulation of the communication within an abstract message passing-based subsystem favors a restructuring or redistribution of the software within the client-server model. Single JAVA classes can even be substituted at run time.

5.4.2 Portability and Integration Ability

The implementation of the system in the programming language JAVA basically enables us to use our software on a diversity of computer systems. This makes load balancing possible even within a cluster of non-uniform machines.

Additionally, by implementing the client in the form of a signed applet, the user interface can rely on the functionality of standard browsers.

The client-server architecture not only does support load balancing, but also provides means for confidently (pre)processing the sample document provided by the user without the necessity to transfer its contents to the unsecure/untrusted server side.

Another important feature is the ability to integrate already existing search systems within our framework. Text based systems can easily use our user interface for text output. Also, existing word stemmers, clustering and indexing mechanisms can be used alternatively. This integration possibility facilitates the user acceptance and helps migrating from already existing older systems to the new one.

5.4.3 Load Balancing

Aside from the already discussed possibilities of migrating some modules between client and server there are a couple of other load distribution possibilities for the modules:

- *parser*
  Beside the possibility of shifting heavy workload of user input processing (e.g. huge PostScript or PDF files) to the client also simple round robin schemes can easily be implemented in a cluster.

- *cluster control*
  The clustering of the index terms (fusion and division of clusters) is one of the most critical workloads. Especially the root node of the classification tree is a heavily frequented data structure which should be mirrored by other computers of the cluster. Low level nodes, e.g. leafs, are not frequently visited and can therefore be moved to the machines with low workload. Between these extremes, common strategies can be used to decide whether or not to move nodes.

- *repository*
  The most simple load balancing strategy consists of the use of several independent repositories for partial hierarchies. Shifting the representatives from one level to another or between nodes on the same hierarchy level might require additional data transfer efforts between the computers who store the affected nodes, i.e. directory and context information/links have to be adjusted. Similar to the parser, a *round robin* load balancing scheme can be implemented, but must be backed up by corresponding data replication.

6 Discussion and Outlook

After the review of state-of-the-art information retrieval concepts and related algorithms we focused on the case of document search on the Internet. We introduced a system architecture and the components of a new adaptive, intuitive Internet document navigation system.

Although our system builds upon the experiences of existing systems, there are many open questions left for our special design. Especially those related to the user interface have to be evaluated in practice.

- *adaptive classification*
  Usage of standard search engines has already been evaluated (Koenemann and Belkin 1996), even for Web visualization (Heo and Hirtle 2001), but truly searchable adaptive directory structures are new. Are users willing to accept structural changes? What are the optimal stability/plasticity parameters?

- *user interface*
  The zoomable interface paradigm is quite new and has not yet been evaluated within an adaptive setting. Does it help the user or does it rather hinder the information retrieval process? This "intuitive" approach might be misleading; perhaps zooming is the wrong metaphor for such a task.

In order to answer these questions, an evaluation stage is planned in cooperation with the German National Library.

Beside these basic questions, there is still much work left:

- *linguistic analysis*
  The "semantic" meaning of identified clusters might be greatly improved if commercial (language-dependent) dictionaries/ thesauri can be used to support the classification of terms (Liddy

1994). Aside from arising licensing problems (client-side preprocessing of documents would certainly be hindered), it is not clear how to match the dictionaries' underlying *static* classifications with the dynamic ones which are generated by our system.

- *content management*
  A content management system deals with the task of management of data formats, data conversion, version control, protocols for web publishing and so on. Our system does not contain these features (yet) as our current focus really is on *content-oriented navigation.* Nevertheless, as stated above, thanks to the modular architecture, this kind of functionality could be added later.

- *coupling of independent systems*
  If there are two (ore more) independent systems,. each one using its own document database, a combination of the hierarchy trees might result in better search results and exploration possibilities compared with contacting each one of them separately. On the other hand, each system administrator may want to maintain his/her own database and may not be willing to fuse the document collections. What should we do? The answer is a time-limited coupling of the systems. The obvious approach would consist in the usage of some sort of *meta crawler*, but in order to enable the user to truly navigate within the resulting classification forest even across system boundaries, additional information needs to be exchanged on the server side.

  The research topic is: Which (sub)modules within the different systems should communicate with each other? The required coordination has to take place at the same time ordinary search tasks are processed by the coupled systems – a difficult task.

In conclusion, adaptive internet navigation provides a lot of new and user friendly topics for content oriented document search. However, the adaptive plasticity in the data structures also implies new challenges for data consistency and user orientation within the information retrieval process. Our approach will provide new insights into balancing stability vs. plasticity of data structures and visualization.

References

For all URL the date of a valid access is given in brackets [..].

Bederson, B. and Hollan, J.D. (1994), "Pad++: a zooming graphical interface for exploring alternate interface physics," *Proc. ACM UIST'94*, ACM Press.

Bederson, B. and Meyer, J. (1998), "Implementing a zooming user interface: experience building Pad++," *Software: Practice and Experience*, **28**(10): 1101-1135, ISSN 1097-024X. URL: http://www.cs.umd.edu/hcil/pad++/papers/spe-98-padimplementation/spe-98-padimplementation.pdf [2001-08-18].

Belkin, N.J. and Croft, W.B. (1992), "Information filtering and information retrieval: two sides of the same coin?" *Communications of the ACM*, **35**(12): 29-38, ISSN 0001-0782.

Blelloch, G. (1998), "Algorithms in the real world," Berkeley University, Class notes 1997/98. URL: http://www.cs.cmu.edu/afs/cs/project/pscico-guyb/294/class-notes/all/AlgsInRealWorld.ps.gz [2001-03-27].

Boley, D.L. (1997), "Principal direction divisive partitioning," Technical Report TR-97-056, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455. URL: ftp://ftp.cs.umn.edu/dept/users/boley/reports/PDDP.ps.gz [2001-03-27].

Boley, D.L. (1998), "Hierarchical taxonomies using divisive partitioning," Technical Report TR-98-012, Department of Computer

Science and Engineering, University of Minnesota, Minneapolis, MN 55455. URL: ftp://ftp.cs.unm.edu/dept/users/boley/reports/taxonomy.ps.gz [2001-03-27].

Boley, D., Gini, M., Gross, R., Han, E.-H., Hastings, K., Karypis, G., Kumar,V., Mobasher, B. and Moore, H. (1999), "Document categorization and query generation on the World Wide Web using WebACE," *Artificial Intelligence Review*, **13**(5-6): 365-391. URL: http://www-users.cs.umn.edu/~gross/papers/aij.agent.ps [2001-03-27].

Brin, S. and Page, L. (1998), "The anatomy of a large-scale hypertextual web search engine," *Proceedings of the 7th International World Wide Web Conference (WWW7)*, Brisbane, Australia. URL: http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm [2001-06-12].

Can, F. and Ozkarahan, E.A. (1983), "A clustering scheme," in Kuehn, J.J. (ed.), *Proceedings of the 6th Annual Int. ACM/SIGIR Conf. on Research and Development in Information Retrieval*, vol. 17, no. 4, ACM Press, Bethesda, Maryland, USA, pp. 115-121, ISBN 0-89791-107-5.

Can, F. and Ozkarahan, E.A. (1984), "Two partitioning type clustering algorithms," *Journal of the American Society for Information Science*, **35**(5): 268-276, John Wiley & Sons, Inc., ISSN 0002-8231.

Can, F. and Ozkarahan, E.A. (1985), "Similarity and stability analysis of the two partitioning type clustering algorithms," *Journal of the American Society for Information Science*, **36**(1): 3-14, John Wiley & Sons, Inc., ISSN 0002-8231.

Can, F. and Ozkarahan, E.A. (1989), "Dynamic cluster maintenance," *Information Processing & Management*, **25**(3): 275-291, Pergamon Press Ltd., ISSN 0306-4573.

Carmen (1999), "CARMEN: Content Analysis, Retrieval and Metadata: Effective Networking." URL: http://www.mathematik.uni-osnabrueck.de/projects/carmen/ [2001-08-17].

Caumanns, J. (1998), "A fast and simple stemming algorithm," Freie Universität Berlin, CeDiS. URL: http://www.wiwiss.fu-berlin.de/~caumanns/i4/papers/se/stemming.ps [2001-02-01].

Cohen, J.D. (1995), "Highlights: language- and domain-independent automatic indexing terms for abstracting," *Journal of the American Society for Information Science*, **46**(3): 162-174, John Wiley & Sons, Inc., ISSN 0002-8231. URL: http://www3.interscience.wiley.com/cgi-bin/fulltext?ID=10050162&PLACEBO=IE.pdf [2001-06-15]. (Erratum in *JASIS* **47**(3): 260. URL: http://www3.interscience.wiley.com/cgi-bin/fulltext?ID=57719&PLACEBO=IE.pdf [2001-06-15].)

Cohen, J.D. (1997a), "Drawing graphs to convey proximity: an incremental arrangement method," *ACM Transactions on Computer-Human Interaction*, vol. 4, no. 3, ACM Press, pp. 197-229. URL: http://www.acm.org/pubs/citations/journals/tochi/1997-4-3/p197-cohen/ [2001-06-08].

Cohen, J.D. (1997b), "Recursive hashing functions for N-grams," *ACM Transactions on Information Systems*, vol. 15, no. 3, ACM Press, pp. 291-320. URL: http://www.acm.org/pubs/citations/journals/tois/1997-15-3/p291-cohen/ [2001-06-08].

Damashek, M. (1995), "Gauging similarity via N-grams: language-independent sorting, categorization, and retrieval of text," *Science*, **267**: 843-848, American Association for the Advancement of Science. URL: http://gnowledge.sourceforge.net/damashek-ngrams.pdf [2001-05-11].

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. (1990), "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, **41**(6): 391-407, John Wiley & Sons, Inc., ISSN 0002-8231. URL: http://www3.interscience.wiley.com/cgi-bin/fulltext?ID=10049585&PLACEBO=IE.pdf [2001-03-29].

Dublin Core Metadata Initiative (1995). URL: http://www.dublincore.org [2001-08-21].

Faloutsos, C. (1992), *Signature Files*, in (Frakes and Baeza-Yates 1992), Chapter 4, pp. 44-65.

Faloutsos, C. and Lin, K.-I. (1995), "FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," *Proceedings of the 1995 Int. ACM/SIGMOD Conf. on Management of Data*, vol. 24, no. 2, ACM Press, pp. 163-174. URL: http://www.acm.org/pubs/ /citations/proceedings /mod/223784/p163-faloutsos/ [2001-04-05].

Fellbaum, C. (ed.) (1998), *WordNet: an Electronic Lexical Database*, MIT Press, Cambridge, Massachusetts [and others], ISBN 0-262-06197-X. URL: http://www.cogsci.princeton.edu/~wn [2001-03-01].

Frakes, W.B. (1992), *Stemming Algorithms*, in (Frakes and Baeza-Yates 1992), Chapter 8, pp. 131-160.

Frakes, W.B. and Baeza-Yates, R.S. (eds.) (1992), *Information Retrieval: Data Structures and Algorithms*, first ed., PTR Prentice-Hall, Inc., Eaglewood Cliffs, New Jersey 07632, ISBN 0-13-463837-9.

Giles, C.L., Bollacker, K.D., and Lawrence, S. (1998), "CiteSeer: an automatic citation indexing system," in Witten, I., Akscyn, R., and Shipman III, F.M. (eds.), *Third ACM Conference on Digital Libraries*, ACM Press, New York, pp. 89-98, ISBN 0-

8979-1965-3. URL: http://www.neci.nj.nec.com/homepages /lawrence/papers/cs-dl98/cs-dl98-letter.pdf [2000-12-15].

Harman, D. (1992), "Relevance feedback and other query modification techniques," in (Frakes and Baeza-Yates 1992), Chapter 11, pp. 241-263.

Heo, M. and Hirtle, S. (2001), "An empirical comparison of visualization tools to assist information retrieval on the Web," *Journal of the American Society for Information Science and Technology*, **52**(8): 666-675, John Wiley & Sons, Inc., ISSN 1532-2882. URL: http://www3.interscience.wiley.com/cgi-bin/fulltext ?ID=80002501&PLACEBO=IE.pdf [2001-06-08].

Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1996), "Newsgroup exploration with WEBSOM method and browsing interface," Report A32, Helsinki University of Technology, Faculty of Information Technology, Laboratory of Computer and Information Science, Rakentajanaukio 2C, SF-02150 Espoo, Finland, ISBN 951-22-2949-8. URL: http://websom.hut.fi/websom/doc/ps /honkela96tr.ps.gz [2001-04-05].

Jones, K.S. and Willett, P. (eds.) (1997), *Readings in Information Retrieval*, The Morgan Kaufmann Series in Multimedia Information and Systems, first ed., Morgan Kaufmann Publishers, San Francisco, CA 94104-3205, ISBN 1-55860-454-5.

Kaszkiel, M. and Zobel, J. (2001), "Effective ranking with arbitrary passages," *Journal of the American Society for Information Science and Technology*, **52**(4): 344-364, John Wiley & Sons, Inc., ISSN 1532-2882. URL: http://www3.interscience.wiley.com/cgi-bin/fulltext?ID=76508338&PLACEBO=IE.pdf [2001-04-30].

Kleinberg, J.M. (1998), "Authoritative sources in a hyperlinked environment," *Proceedings of the 9th ACM/SIAM Symposium on Discrete Algorithms*, ACM Press, pp. 668-677. (Extended version appeared in *Journal of the ACM* **46**(5): 604-632.) URL: http://www.acm.org/pubs/articles/journals/jacm/1999-46-5/p604-kleinberg/p604-kleinberg.pdf [2001-06-16].

Koenemann, J. und Belkin, N.J. (1996), "A case for interaction: a study of interactive information retrieval behaviour and effectiveness," in Bilger, R., Guest, S., and Tauber, M.J. (eds.), *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems*, ACM Press, Vancouver, BC, Canada, pp. 205-212, ISBN 0-89791-777-4. URL: http://www.acm.org/pubs/citations/proceedings/chi/238386/p205-koenemann/ [2001-03-29].

Kruskal, J.B. (1964a), "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, pp. 1-27.

Kruskal, J.B. (1964b), "Nonmetric multidimensional scaling: a numerical method," *Psychometrika*, vol. 29, pp. 115-129.

Leuski, A. (2000a), "Details of Lighthouse," Technical Report IR-212, Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, Amherst, MA 01003. (Extended version of (Leuski 2000b).) URL: http://ciir.cs.umass.edu/pubfiles/ir-212.pdf [2001-03-26].

Leuski, A. (2000b), "Relevance and reinforcement in interactive browsing," *Proceedings of the Ninth* International *Conference on Information and Knowledge Management (CIKM)*, Washington, DC. (Also available as CIIR Technical Report IR-208, Department of Computer Science, University of Massachusetts, Amherst, MA 01003.) URL: http://ciir.cs.umass.edu /pubfiles/ir-208.pdf [2001-03-26].

Liddy, E.D. (1994), "Text categorization for multiple users based on semantic features from a machine-readable dictionary," *ACM Transactions on Information Systems*, **12**(3): 278-295, ACM

Press. URL: http://www.acm.org/pubs/citations/journals/tois/1994-12-3/p278-liddy/ [2001-03-26].

Luhn, H.P. (1958), "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, **2**: 159-165, International Business Machines Corporation, ISSN 0018-8646.

Meghabghab, G. (2001), "Google's web page ranking applied to different topological web graph structures," *Journal of the American Society for Information Science and Technology*, **52**, John Wiley & Sons, Inc., ISSN 1532-2882. URL: http://www3.interscience.wiley.com/cgi-bin/fulltext?ID=82002488&PLACEBO=IE.pdf [2001-06-15].

Moffat, A. and Zobel, J. (1996), "Self-indexing inverted files for fast text retrieval," *ACM Transactions on Information Systems*, vol. 14, no. 4, ACM Press, pp. 349-379. URL: http://www.acm.org/pubs/citations/journals/tois/1996-14-4/p349-moffat/ [2001-04-05].

Open Directory Project. URL: http://dmoz.org [2001-09-01].

Perlin, K. and Fox, D. (1993), "Pad – an alternative approach to the computer interface," *Proceedings of the ACM SIGGRAPH Conference*, vol. 28, ACM Press, Anaheim, USA, pp. 57-64. URL: http://www.cs.umd.edu/hcil/pad++/papers/siggraph-93-origpad/siggraph-93-origpad.ps.gz [2001-08-18].

Porter, M.F. (1980), "An algorithm for suffix stripping," *Program*, **14**(3): 130-137, reprinted in (Frakes 1992). URL: http://www.tartarus.org/˜martin/PorterStemmer/ [2001-06-15].

Raskin, J. (2000), *The* Humane *Interface: New Directions for Designing Interactive Systems*, first ed., Addison Welsley Longman, Inc., Reading, Massachusetts 01867, ISBN 0-2-1-37937-6.

Rasmussen, E. (1992), "Clustering algorithms," in (Frakes and Baeza-Yates 1992), Chapter 16, pp. 419-442.

Sahami, M. (1998), *Using Machine Learning to improve Information Access*, PhD thesis, Stanford University, Department of Computer Science. URL: http://robotics.stanford.edu /users/sahami/papers-dir/thesis.ps [2001-07-09].

Salton, G. and Buckley, C. (1988), "Term weighting approaches in automatic text retrieval," *Information Processing & Management*, **24**(5): 513–523, Pergamon Press Ltd., ISSN 0306-4573. (Also available as Technical Report 87-881, Cornell University, Department of Computer Science, Ithaca, New York 14853.) URL: http://cs-tr.cs.cornell.edu/Dienst/UI/1.0/Display/ncstrl.cornell /TR87-881/ [2001-04-05].

Salton, G., Wong, A., and Yang, C.S. (1975), "A vector space model for automatic indexing," *Communications of the ACM*, **18**(11): 613-620, ISSN 0001-0782. (Also available as Technical Report 74-218, Cornell University, Department of Computer Science, Ithaca, New York 14853.) URL: http://cs-tr.cs.cornell.edu /Dienst/UI/1.0/Display/ncstrl.cornell/TR74-218/ [2001-04-05].

Salton, G., Wong, A., and Yu, C.T. (1976), "Automatic indexing using term discrimination and term precision measurements," *Information Processing & Management*, **12**: 43-51, Pergamon Press Ltd., ISSN 0306-4573.

Salton, G., Wu, H., and Yu, C.T. (1981), "The measurement of term importance in automatic indexing," *Journal of the American Society for Information Science*, **32**(3): 175-186, John Wiley & Sons, Inc., ISSN 0002-8231.

Topic Maps: XML schema of ISO 13250. URL: http://www.diffuse .org/TopicMaps/schema.html [2001-09-29].

Ueberall, M. (2001), "Ein adaptives, hierarchisch organisiertes, verteiltes Navigationssystem zur inhaltsbasierten Dokumentensuche," Diplom Thesis, Institut für Informatik, Johann Wolfgang Goethe-Universität, Frankfurt/Main, Germany.

van Rijsbergen, C.J. (1979), *Information Retrieval*, second ed., Butterworths, London, ISBN 0-408-70929-4. URL: http://www.dcs.gla.ac.uk/Keith/Preface.html [2001-07-12].

Vivísimo. URL: http://www.vivisimo.com [2001-09-17]

Wätjen, H.-J., Diekmann, B., Möller, G., and Carstensen, K.-U. (1998), "Bericht zum DFG-Projekt GERHARD: German Harvest Automated Retrieval and Directory," Technical Report, Bibliotheks- und Informationssystem (BIS), Carl von Ossietzky Universität Oldenburg. URL: http://www.gerhard.de/info/dokumente/dokumentation/gerhard/bericht.pdf [2001-07-23].

WebMap. URL: http://www.webmap.com [2001-09-17]

Willett, P. (1988), "Recent trends in hierarchic document clustering: a critical review," *Information Processing & Management*, **24**(5): 577-597, Pergamon Press Ltd., ISSN 0306-4573.

Wong, S.K.M., Ziarko, W., and Wong, P.C.N. (1985), "Generalized vector space model in information retrieval," *Proceedings of the 8th Annual Int. ACM/SIGIR Conf. on Research and Development in Information Retrieva*l, ACM Press, Montreal, Canada, pp. 18-25. URL: http://www.acm.org/pubs/citations/proceedings/ir/253495/p18-wong/ [2001-04-13].

Wulfekuhler, M.R. and Punch, W.F. (1997), "Finding salient features for personal Web page categories," *Proceedings of the Sixth International World Wide Web Conference*. URL: http://www.cps.msu.edu/~wulfekuh/research/PAPER118.ps [2001-07-23].

Xu, J. and Croft, W.B. (1998), "Corpus-based stemming using co-occurence of word variants," *ACM Transactions On Information Systems*, vol. 16, no. 1. (Also available as CIIR Technical Report IR-95, Department of Computer Science, University of Massachusetts, Amherst, MA 01003.) URL: http://www.acm.org /pubs/citations/journals/tois/1998-16-1/p61-xu/ [2001-06-08].

Zhang, J. and Korfhage, R.R. (1999), "A distance and angle similarity measure method," *Journal of the American Society for Information Science*, **50**(9): 772-778, John Wiley & Sons, Inc., ISSN 0002-8231. URL: http://www3.interscience.wiley.com/cgi-bin/fulltext?ID=10050162&PLACEBO=IE.pdf [2001-06-12].

Zhang, J. and Wolfram, D. (2001), "Visualization of term discrimination analysis," *Journal of the American Society for Information Science and Technology*, **52**(8): 615-627, John Wiley & Sons, Inc., ISSN 1532-2882. URL: http://www3.interscience .wiley.com/cgi-bin/fulltext?ID=79502836&PLACEBO=IE.pdf [2001-06-08].

Zipf, G.K. (1949), *Human Behaviour and the Principle of Least Effort – an Introduction to Human Ecology*, Addison-Wesley, Cambridge, Massachusetts, ISBN 0-012-78978-1. (Facsimile: Hafner Publishing Company, New York, 1965.)

Zobel, J. and Moffat, A. (1998), "Exploring the similarity space," *ACM SIGIR Forum*, **32**(1): 18-32, ACM Press. URL: http://www .cs.mu.oz.au/~alistair/abstracts/zm98:forum.html [2001-06-08].

This page is intentionally left blank

Chapter 3

Flexible Queries to XML Information

E. Damiani, N. Lavarini, S. Marrara, B. Oliboni, and L. Tanca

The *eXtensible Markup Language* (XML) (W3C 1998a) was initially proposed as a standard mark-up language for representing, exchanging and publishing information on the Web. It has recently spread to many other application fields: to name but a few, XML is currently used for multi-protocol Web publishing, for legacy data revamping, for storing data that cannot be represented with traditional data models and for ensuring inter-operability among different software systems and applications. The *HyperText Markup Language* (HTML) itself has recently been re-defined in terms of XML. Furthermore, XML information is often stored and transmitted in text form (though other binary serialization formats are also available); therefore, the availability of standard character-based encodings for text is making XML the solution of choice for long term storage of slow-obsolescence data. For the sake of conciseness, in this chapter we shall provide only a skeletal introduction to XML (the interested reader is referred to the standard documentation (W3C 1998a) or to the book (Box 2001)).

1 A Concise Introduction to XML

Generally speaking, an XML dataset (usually called *document*) is composed of a sequence of nested elements, each delimited by a pair of start and end tags (e.g., `<tag>` and `</tag>`). XML documents can be broadly classified into two categories: *well-formed* and *valid*. An XML document is *well-formed* if it obeys the basic syntax of

XML (i.e., non-empty tags must be properly nested, each non-empty start tag must have the corresponding end tag). The sample well-formed XML document that will be used throughout the chapter is shown in Figure 1, while its Infoset tree is pictorially represented in Figure 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<computer>
  <maker> Toshiba </maker>
  <model serialcode = “12303B”>
    <modelname> Satellite Pro 4200 </modelname>
    <year> 2001 </year>
    <description>
      A versatile laptop computer product.
    </description>
  </model>
  <plant>
    <address> Osaka, Japan</address>
  </plant>
</computer>
```

Figure 1. A well-formed XML document.

The Infoset defines three *content models* for XML elements: the *element-only* content model, which allows an XML tag to include other elements and/or attributes, but no text, the *text-only* content model, allowing a tag to contain text and attributes only, and the *mixed* model, allowing a tag to contain both sub-elements and text. The latter model, though still used in some document processing applications, is deprecated for XML-based formats in other domains.

Well-formed XML documents are also *valid* if they conform to a proper *Document Type Definition* (DTD) or XML Schema. A DTD is a file (external, included directly in the XML document, or both) which contains a formal definition of a particular type of XML documents. Indeed, DTDs include declarations for *elements* (i.e. tags), *attributes*, *entities*, and *notations* that will appear in XML documents. DTDs state what names can be used for element types, where

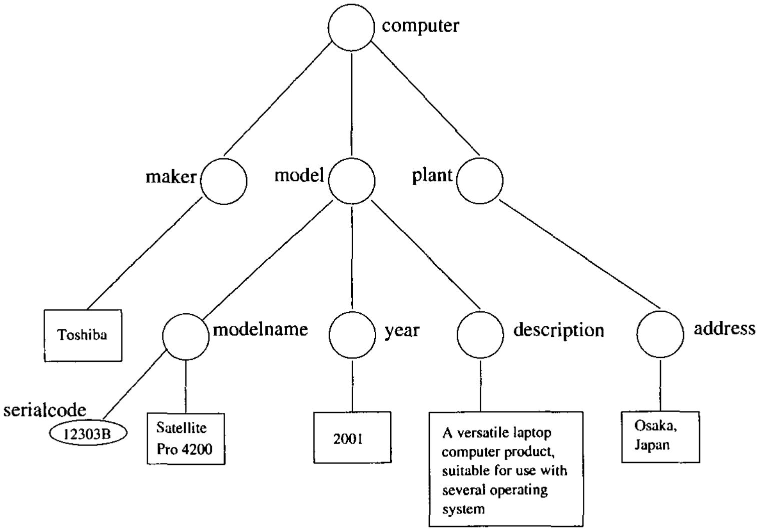


Figure 2. The Infoset tree for the document of Figure 1.

they may occur, how each element relates to the others, and what attributes and sub-elements each element may have. Attribute declarations in DTDs specify the attributes of each element, indicating their name, type, and, possibly, default value. A sample DTD for the document of Figure 1 is reported in Figure 3.

```
<!ELEMENT address (#PCDATA)>
<!ELEMENT computer (maker, model, plant)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT maker (#PCDATA)>
<!ELEMENT model (modelname, year, description)>
<!ATTLIST model serialcode CDATA #REQUIRED>
<!ELEMENT modelname (#PCDATA)>
<!ELEMENT plant (address)>
<!ELEMENT year (#PCDATA)>
```

Figure 3. A sample DTD.

Note that, due to the semi-structured nature of XML data, it is possible (and, indeed, frequent) for two instances of the same DTD to have a different structure. In fact, some elements in the DTD can be optional and other elements can be included in an XML document zero, one, or multiple times. While it is usually substantially longer than a DTD, an XML Schema definition carries much more information, inasmuch it allows the document designer to define XML data structures starting from reusable *simple and complex types* and then declaring XML elements as variables belonging to those types. A sample schema for the document of Figure 1 is reported in Figure 4. Note that elementary types (such as strings, integers and the like) need not be explicitly defined as they are part of the XML Schema standard (denoted in our sample document by the prefix `xsd:`). On the other hand, types `modelType` and `plantType` are explicitly defined in Figure 4 and then elements `<model>` and `<plant>` are declared as belonging to those types. XML Schema is currently the solution of choice adopted by software designers for defining XML-based formats for information interchange on the Internet, while DTDs are still widely used by the document management community; however, we remark that the XML standard data model, called *Infoset* is not a text-based one, as it represents both XML schemata and documents as *multi-sorted trees*, i.e. trees including nodes belonging to a variety of types. The *Document Object Model* (DOM) Level 2 standard (W3C 1998b) defines a lower level, programming language independent *application program interface* to such trees. While the XML Infoset defines several types of nodes, in this chapter, for the sake of clarity, we shall focus on *Element* and *Attribute* nodes, corresponding to XML tags and attribute-value pairs, and on *Text* nodes, corresponding to XML tags' content. For our purposes, an XML tree including such node types (Figure 2) fully defines an XML document's structure and its content.

The *validation* or syntax-checking procedure is performed by a *XML validating parser* and involves a well-formed XML document and a DTD or XML Schema: if the XML document is valid with respect to the DTD or Schema, the validating parser usually produces a tree-

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10
  /XMLSchema" elementFormDefault="qualified">
  <xsd:element name="address" type="xsd:string"/>
  <xsd:element name="computer">
    <xsd:complexType> <xsd:sequence>
      <xsd:element ref="maker"/>
      <xsd:element name="model" type="modelType"/>
      <xsd:element name="plant" type="plantType"/>
      </xsd:sequence> </xsd:complexType>
  </xsd:element>
  <xsd:element name="description" type="xsd:string"/>
  <xsd:element name="maker" type="xsd:string"/>
    <xsd:complexType name="modelType">
      <xsd:sequence>
        <xsd:element ref="modelname"/>
        <xsd:element ref="year"/>
        <xsd:element ref="description"/>
      </xsd:sequence>
  <xsd:attribute name="serialcode" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:binary">
      <xsd:encoding value="hex"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  </xsd:complexType>
    <xsd:element name="modelname" type="xsd:string"/>
  <xsd:complexType name="plantType">
    <xsd:sequence>
      <xsd:element ref="address"/>
    </xsd:sequence>
  </xsd:complexType>
    <xsd:element name="year" type="xsd:short"/>
</xsd:schema>
```

Figure 4. A sample XML schema.

shaped memory representation of the document according to a lower-level application program interface, such as the *Document Object Model* (DOM) Level 2 standard.

1.1 Dealing with Heterogeneous, not Validated XML Data

While many current XML processing techniques rely on DTDs or Schemata, it is important to remark that not all XML datasets used in practice actually comply with a DTD or a Schema. Currently, a large amount of XML information is being made available on the WWW in unvalidated text form; also, XML data coming from heterogeneous data sources, while carrying similar semantics, may comply with DTDs or schemata which are themselves heterogeneous or simply unavailable. Figure 5 shows a well-formed document that, while carrying more or less the same semantics as the one in Figure 1, complies with a very different Schema (or DTD).

```
<?xml version="1.0" encoding="UTF-8"?>
<computer>
  <description>
    A versatile laptop computer product.
  </description>
  <maker> Toshiba </maker>
  <model serialcode=“12303B” modelname=“Satellite Pro4200”
    year=“2001”>
  </model>
  <plant address="Osaka" country="Japan"/>
</computer>
```

Figure 5. A semantically equivalent, but syntactically different XML document.

As we shall see in Section 2, most current query and processing languages for XML rely on the notion of a well-known, single schema or DTD underlying all XML data to be queried. Therefore, XML query languages do not fully address the need of Web-enabled applications to access, process and query heterogeneous XML documents, flexibly dealing with variations in their structure. Our approach to flexible query techniques will be described in detail in the remainder of the chapter.

2 The Problem of Flexible Queries

Researchers have proposed many different techniques (e.g., algebraic ones (Clarke *et al.* 1995)) for searching and structuring XML documents. In the last few years, the database community has proposed several fully-fledged query languages for XML, some of them as a development of previous languages for querying semi-structured data; two detailed comparisons (both involving four languages) can be found in (Ceri and Bonifati 2000) and (Cluet *et al.* 1999a), while many preliminary contributions and position papers about XML querying are collected in (W3C 1998c). Although we shall provide some query examples, in the following we do not attempt to describe XML query languages in detail; rather, we shall only refer to the common features of XQL (Robie 1999), XML-QL (Deutsch *et al.* 1999), YaTL, (Robie *et al.* 2000), XML-GL (Ceri *et al.* 1999) and the recent XQuery standard (Robie 1999). Specifically, two features shared by all these languages (Cluet *et al.* 1999a, Robie *et al.* 2000) are relevant to our discussion:

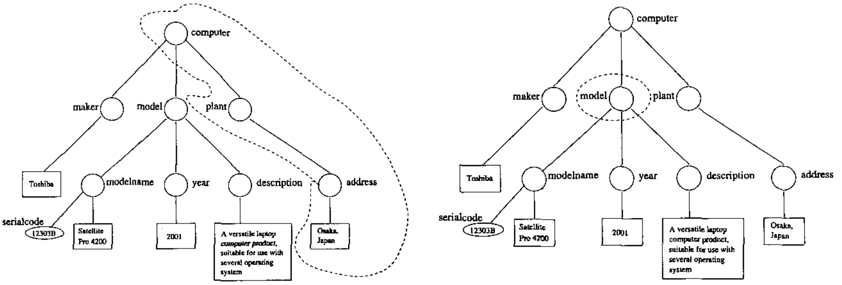
- Query are based on *User-provided patterns*, relying on the assumption that the user is aware enough of the target document structure to be able to formulate a pattern that can be matched against the target XML documents for locating the desired information, producing a *node-set* matching the pattern. Syntactically, all XML query languages (and, in particular, XQuery) express patterns via suitable extensions to a standard form based on *XPaths* (Cluet *et al.* 1999a). XPaths have since long been adopted as a W3C Recommendation and, besides being used in the framework of XML query languages, have been incorporated into several other XML-related standards such as XPointer and XSLT. In all these contexts, XPath expressions have a role not unlike the

SELECT clause of a SQL statement. Each pattern represents a series of steps in the target document along a specific *axis*. XPath axes allow the user to specify path expressions that follow different *orders of visit* in the target tree; the *child* axis corresponds to the intuitive order from root to leaves. For instance, the XPath `/computer/model/*` (along the child axis) will select the nodes `<modelname>` and `<year>` in Figure 1, together with their content. Generally speaking, XPaths can be seen as a special case of general *tree patterns* whose application to a target document returns a *forest* of nodes, preserving their hierarchy and sequence. As the previous example suggests, flexibility support is obtained in XPath by means of wildcards. This feature is also shared by the XQL language, though its pattern syntax does not exactly coincide with XPath (Robie 1999).

- *Set-oriented query result*: all query languages relying on XPath-like patterns for selecting node sets retrieve portions of XML documents, namely the ones fully matching the user-provided pattern, and enrich them with new information with the help of various constructor. Although XML query languages use different binding techniques (Robie *et al.* 2000), they share a common feature: all retrieved portions equally belong to the query result set, even when the query exploits the facilities provided by the language for partial or flexible pattern matching.

2.1 XML Query Languages and Flexible Search

With respect to the first feature listed in Section 2 we observe that, when querying large amounts of heterogeneous XML information, the assumption that the user is aware of the target document structure is indeed debatable, because users cannot exploit a DTD or a XML Schema as a basis for the query structure. Often, all users can rely on a set of sample documents, or at most a tag repertoire, i.e. the XML *vocabulary* used throughout the XML document base. In this situation, trying to find a match of the query to a part of the target document is likely to result in *silence*, as the query topology, however similar, will probably not exactly match the document's structure. In order to further clarify this point, consider the simple XQL query `//Computer[Plant contains "Japan"] (.//Model )`. The syntax of this query (whose syntax could be easily translated into XQuery) is quite self-explanatory: the first part of the query uses an XPath-like expression (namely, `//Computer[Plant contains "Japan"]`) to identify all XML subtrees rooted in a `<computer>` node containing a `<plant>` child node whose content is "`Japan`". The second part defines what should be retrieved, i.e. the whole `<model>` node of the selected subtree. It is easy to see that when applied to the document of Figure 1, the query will identify the XML subtree contained in the dashed region of Figure 6a and will indeed output the `<model>` node as shown in Figure 6b, while if the document of Figure 5 is used, the empty node-set (i.e., silence) will result. In other words, even slight variations in datasets' structures may result in unexpected silence in the query answer. Once again, we remark that when heterogeneous data sources are involved, queries rarely intend to dictate the exact structure of the query result; rather, they provide a loose example of the information the user is interested in. Therefore several degrees of matching should be possible. This situation has been dealt with in the field of multi-media databases (Fagin 1996) where query results are typically ranked lists according to some similarity measure.



(a) The retrieved subtree (b) The result node

Figure 6.

2.2 The Role of Text Retrieval Techniques

When XML Schema or DTD information is not available for a given XML dataset, one might think of taking advantage of the text-based serialization format used for most XML information. The fact that most XML data are available as plain text suggests to use standard text retrieval techniques, like the ones currently exploited by WWW search engines. At first sight, this solution may look appealing as standard text retrieval techniques could well be used to search for tags (such as, in the sample document of Figure 1, `<model>` and `<address>`) as well as for their desired content. We briefly recall that standard *Boolean* techniques for text retrieval search rely on a *lexicon*, i.e. a set of terms $r_1, r_2, .., r_k$ and model each document as a Boolean vector of length k, whose i-th entry is `true` if r_i belongs to the document. In this setting, a query is simply a Boolean expression (e.g., a conjunction) whose operands are terms or stems (possibly including *wildcards*), and its result is the set of documents where the Boolean expression evaluates to `true`. In other words, though a variety of fuzzy techniques have been proposed to overcome this problem (Radecki 1979) document ranking is not supported in a pure Boolean setting. On the other hand, *probabilistic* text retrieval techniques model documents as *multisets* of terms, and queries as standard sets of terms, aiming at computing $P(R/Q, d)$, i.e. the probability that a document d is relevant with respect to query Q, based on the frequency distribution of terms inside d. The result is usually a ranked list of documents according to values of $P(R/Q, d)$. Variations of these techniques are currently in use for search engines dealing with HTML documents, and could of course be employed for XML data as well, but at the price of loosing all the information conveyed by the document's structure. This loss is indeed very important when the XML elements' content is made of typed values rather than of text *blobs* (as in the sample document of Figure 1), as it is nearly always the case when XML documents are dynamically extracted from relational databases. Finally, it should be noted that, as hinted at in Sect. 1, text is only one of the possible serialization formats of XML, and several binary serializations are used in various application domains (e.g. in wireless communication). Therefore, the assumption that XML is encoded as plain text is not always valid and should be removed. In the following, we shall abandon it in favor of a weaker one, assuming that the target XML data serialization format is any format that:

- complies with the XML Infoset
- can be parsed towards a low-level representation accessible via the DOM application program interface.

3 The ApproXML Approach: an Outline

Our approach to flexible XML querying (Damiani and Tanca 2000) is based on a flexible search and processing technique, capable of extracting relevant information from a (possibly huge) set of heterogeneous XML documents. Such XML documents may come from a number of different data sources, each employing its own mark-up; this corresponds to a high degree of variability about the documents' structure and tag vocabulary. The relevant information is selected as follows:

- first, an *XML pattern*, i.e. a partially specified subtree, is provided by the user;
- then, the arcs of the target XML documents are weighted and their structure is extended by adding new arcs in order to by-pass links and intermediate elements which are not relevant from the user's point of view;
- finally, extended documents are scanned and matching XML fragments are located and sorted according to their similarity to the pattern.

This pattern-based approach is a step forward with respect to keyword-based solutions currently used by WWW search engines, as the latter cannot exploit XML mark-up, and therefore ignore potentially useful information about document structure. As we shall see, the problem of matching the pattern to the extended XML tree structure can be described as a *fuzzy sub-graph matching* problem. A sample fuzzy tree is shown in Figure 7.

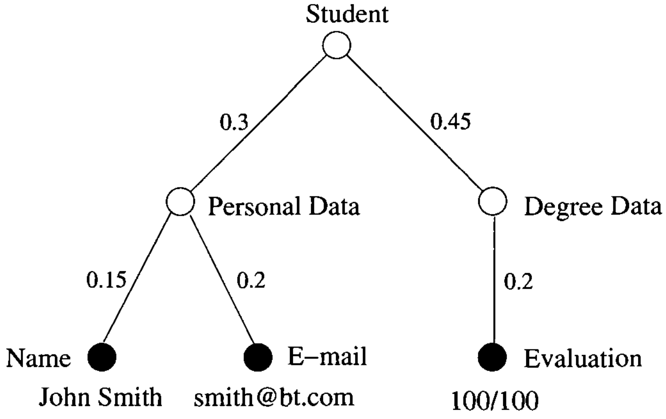


Figure 7. A fuzzy tree.

Besides dealing with structure, however, any approximate matching technique needs to assess the degree of equivalence of the information carried by two nodes or subtrees (Damiani *et al.* 2001). Such hidden equivalence should be reconciled by means of flexible and approximate *smushing* techniques[1]. The notion of *node smushing* can be best understood via an example: suppose that one node of an XML tree contains a direct link to a resource, while another node of a different tree contains an indirect link (with a sequence of pointers) to the same resource. Alternatively, suppose that two nodes having different tags differ only in a small portion of their attributes/content. Obviously, such node pairs are not crisply equivalent, but still they should be treated as equivalent to a degree, i.e. fuzzily. When we ac-

[1] For the original definition of the smushing problem by Dan Brinkley, see http://lists.w3.org/Archives/Public/www-rdf-interest/2000Dec/0191.html

cept such notion of node equivalence, we say that we *smush* the two nodes together.

4 The ApproXML Approach in Detail

We are now ready to describe ApproXML operation in some detail. Intuitively, we want to match the query pattern against the document only after having extended the XML document's tree in order to bypass links and intermediate elements which are not relevant from the user's point of view. In order to perform the extension in a sensible way, we shall first evaluate the *importance* of well-formed XML information at the granularity of XML elements. To achieve this result, we rely on *fuzzy weights* to express the *relative importance* (Bosc 1998) of information at the granularity of XML elements and attributes.

4.1 Weight Computation

Our weights are attributed to arcs; this choice allows for flexibly dealing with the common practice of *tags reuse*, where the same tag may appear in different locations in the same DTD or Schema[2]. Furthermore, our weights monotonically take values in the unit interval; values close to 0 will correspond to a negligible amount of information, while a value of 1 means that the information provided by the element (including its position in the document graph) is extremely important according to the document author. Other than that, the semantics of our weights is only defined in relation to other weights in the same XML document. Of course, we would like the computation of such weights to be carried out automatically, or at least to require limited manual effort. However, the topic of automatically weighting all the arcs of a document graph is quite complex. An error-free importance attribution can be made only "by hand" and,

[2] Tag reuse has nothing to do with the obvious fact that the same tag can (and usually does) appear multiple times inside the same XML *document*.

even then, the task becomes harder as the depth and branching level of the XML tree increases. However, since manual weighting obviously does not scale up, an automatic weighting procedure ought to be attempted. Different approaches may be used for automatic weighting ; for instance, a weighting method based upon (normalized, inverse-of) distance from the root (Damiani *et al.* 2000) relies on the assumption that XML elements' generality (and thus, probably, importance) grows when getting closer to the root. This assumption looks reasonable remembering that the XML Schema standard encourages a style of types' and elements' definition not unlike the one used in object-oriented class design (Section 1). On the other hand, we must take into account the (fundamental) fact that, in most XML documents, values (i.e. text) are located inside terminal elements, possibly at the maximum depth. This means that, in general, any weighting technique should not give low weights to arcs leading to content-filled elements. Our content-insensitive, automatic arc weighting method takes into account the following factors:

Depth The "closeness" to the root, hinting at the generality of the concepts tied by the arc.

Content The "amount" of PCDATA content, to detect if the arc leads to actual data.

Fan-out The number of nodes directly reachable from the destination node of the arc, to detect whether the arc leads to a wide branching of concepts.

Tag name Though content-insensitive, our technique easily takes into account the presence of particular tag names (possibly by using application-domain specific Thesauri), and increases the weights of arcs incoming to nodes whose names are considered as "content bearers"[3].

Our automatic weighting technique takes separately into account all the factors listed above, generating a value for each of them, and

[3] This factor heavily depends on the reliability of the underlying dictionary. We shall not take it into account in the remainder of the chapter.

then aggregates these values within a single arc-weight. We will now describe how individual values are generated.

4.1.1 Depth

It is quite intuitive that the containment relationship between XML elements causes generality to decrease as depth grows, and so we define a standard decreasing hyperbolic function that gives the lowest weights to the deepest nodes. If $a = (n_1, n_2)$ is an arc then

$$w_d(a) = \frac{\alpha}{\alpha + \mathrm{depth}(n_1)} \tag{1}$$

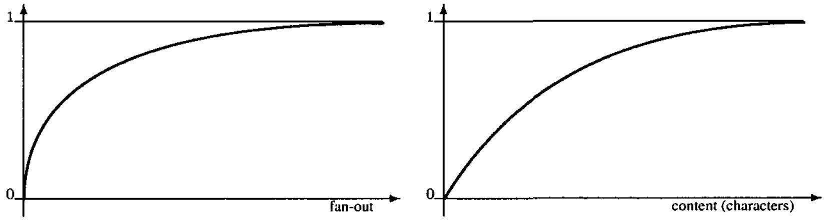
where α can be easily tuned. Let us suppose, as an example, to have a tree with maximum depth 10. It is easy to see that, with $\alpha = 1$ the weights go from 1 to $1/11$, and with $\alpha = 10$ the weights go from 1 to $1/2$. The choice of α can also depend on the value of the maximum depth D. It is easy to show that, in this case, if $\alpha = D/k$ then the minimum weight is $1/(k + 1)$.

4.1.2 Fan-out

The importance of an arc is also given by the number of nodes it leads to. In fact, the fan-out of an arc (n_1, n_2) is defined as the number of elements contained in n_2. To express a value related to fan-out, we believe that, again, a simple function may be used. If $F(a)$ is the fan-out of arc a, we can define

$$w_f(a) = \frac{F(a)}{F(a) + \beta} \tag{2}$$

with $\beta > 0$ to be tuned as needed. In this case, the function obtained is similar to that in Figure $8(a)$. If we assume $\beta = 1$ we have that, if the fan-out of an arc is k, then its weight will be $k/(k+1)$, that tends asymptotically to 1 as k grows.



(a) w_f vs. fan-out (b) w_c vs. content length

Figure 8. Hyperbolic functions.

4.1.3 Content

The techniques described above tend to give to leaf nodes less weight than they deserve, because they often are deep inside the document, and have no children. Indeed, this is a problem because leaf nodes are the main information bearers (especially if the mixed content model for tags is not used (see Section 1)), and giving them a low weight would potentially lead to neglecting useful content. For these reasons, we also take into account XML nodes' *amount of content.* Since our weighting technique is content-insensitive, the only way we have to quantify the amount of information of a node is to calculate the length of its PCDATA string[4].

This means that, given an arc $a = (n_1, n_2)$ its weight (based on its content) should be proportional to the length of the text contained in n_2. If $C(a)$ is the text content of the destination node of a, then

$$w_c(a) = \frac{|C(a)|}{|C(a)| + \gamma} \tag{3}$$

where $|C(a)|$ is the length of $C(a)$, that can be expressed either in tokens (words) or in bytes (characters). As usual, a parameter γ can

[4] Experience shows that this approach is reasonable when the target documents contain a substantial amount of text, while in structured documents, more importance should be attached to the previous two factors.

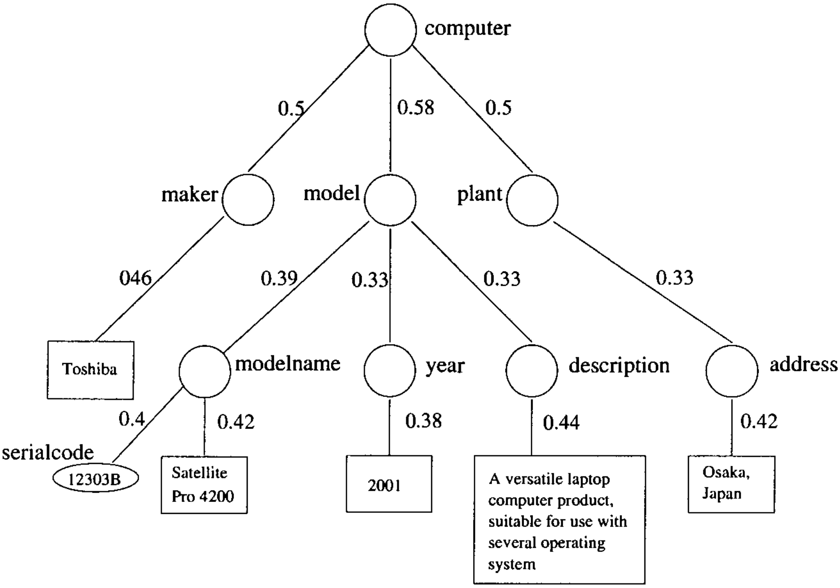


Figure 9. The weighted XML tree corresponding to the document of Figure 1.

be used to tune the slope. Actually, γ represents the content length for which $w_c = 0.5$.

Several techniques (Bardossy *et al.* 1993) can be used to combine weights based on the three factors described above. Aggregation (e.g., based on a weighted average), suggest a *compensatory* vision, where reconciliation of conflicting evaluations is obtained by computing a suitable function of all their values. This function may have different mathematical properties (e.g., it may or may not be linear), and it may take into account any total ordering defined on the factors. In our case, there is no such natural ordering: Figure 9 shows the tree of Figure 2 weighted by aggregating the three factors described above using the simple arithmetic average.

An interesting alternative are *Ordered Weighted Average* (OWA) operators, that provide a linear, order-sensitive aggregation operator often used in information retrieval. It should also be noted that since the semantics of our weights is one of importance evaluation, the different factors take into account potentially conflicting characteristics of the data; therefore, factors reconciliation is not necessarily based on aggregation operators. For instance, *majority-rules* consider weights as (percentage of) votes cast in a ballot (with the additional constraint their sum to be 1), while *tie-break* approaches reconcile potentially conflicting evaluations by using one of them in order to decide when the others disagree. We shall review aggregation techniques in more detail in Section 7.

5 Fuzzy Closure Computation

Once the weighting is completed, the target XML information can be regarded as a *fuzzy tree*, i.e. a tree where all arcs are labeled with a weight taking values in the interval $[0, 1]$. At this point, the *fuzzy transitive closure* C of the fuzzy labeled tree is computed. Intuitively, computing graph-theoretical transitive closure of the XML tree entails inserting a new arc between two nodes if they are connected via a path of any length in the original tree. If the original graph is *directed*, closure may be preserve the direction of its arcs[5]. In general, the complexity of graph closure computation is well-known to be polynomial w.r.t. the number of nodes of the graph. In our model, the weight of each closure arc is computed by aggregating via a t-norm T the weights of the arcs belonging to the path it corresponds to in the original graph. Namely, for each arc (n_i, n_j) in the closure graph C we write:

$$w_{arc}(n_i, n_j) = T(w_{arc}(n_i, n_r), w_{arc}(n_r, n_s), ..., w_{arc}(n_t, n_j)) \quad (4)$$

where $\{(n_i, n_r)(n_r, n_s), ..., (n_t, n_j)\}$ is the set of arcs comprising the shortest paths from n_i to n_j in G and, again, T is a standard t-norm (Klir and Folger 1988). Intuitively, the closure computation

[5] In the following, we shall deal with the undirected case unless otherwise stated.

step gives an extended structure to the document, providing a looser view of the containment relation. Selecting the type of t-norm to be used for combining weights means deciding if and how a low weight on an intermediate element should affect the importance of a nested high-weight element.

This is indeed a time-honored problem and can be very difficult, as the right choice may depend on the specific data set or even on the single data instance at hand. There are some cases in which the t-norm of the minimum best fits the context, other cases in which it is more reasonable to use the product or the Lukasiewicz t-norm. Often, it is convenient to use a family of t-norms indexed by a tunable parameter. In general, however, it is guessing the right context, or better the knowledge associated to it from some background of preliminary knowledge, that leads to the right t-norm for a given application. For instance, suppose a node n_j is connected to the root via a single path of length 2, namely $(n_{root}, n_i)(n_i, n_j)$. If $w_{arc}(n_{root}, n_i) << w_{arc}(n_i, n_j)$ the weight of the closure arc (n_{root}, n_j) will depend on how the t-norm T combines the two weights. In other words, how much should the high weight of (n_i, n_j) be depreciated by the fact that the arc is preceded by (comparatively) low-weight one (n_{root}, n_i)? It is easy to see that we have a conservative choice, namely $T = min$. However, this conservative choice does not always agree with humans' intuition, because the min operator gives a value that depends only on one of the operands without considering the other (Dubois and Prade 1996) (for instance, we have the *absorption property*: $T(x, 0) = 0$). Moreover, it does not provide the *strict-monotonicity* property $(\forall y, x' > x \rightarrow T(x', y) > T(x, y))$. In other words, an increase in one of the operands does not ensure the result to increase if the other operand does not increase as well. To understand the effect of the min's *single operand dependency* in our case, consider the two arc pairs shown below:

```
1. (<computer><model>0.2)
   (<model><serialcode>0.9)
```

```
2.  (<computer><model>0.3)
    (<model><serialcode>0.4)
```

when the min operation is used for conjunction, arc pair (2) is ranked above arc (1), while most people would probably decide that arc pair (1), whose second element has much higher importance, should be ranked first. The other t-operators have the following common properties (Klir and Folger 1988):

$$x = 1 \vee x = 0 \vee y = 1 \vee y = 0 \rightarrow T(x,y) = x \vee T(x,y) = y \quad (5)$$

$$T(x,y) \leq min(x,y) \quad (6)$$

Property 6 warns us that, while the other t-norms somewhat alleviate the single operand dependency problem of the min for arc pairs (using the product, for instance, the outcome of the previous example would be reversed), they may introduce other problems for longer paths. Let's consider the following example, where we add a `modelnamecode` attribute to the `<modelname>` element:

```
1.  (<computer><model>0.1)
    (<model><modelname>0.9)
    (<modelname><modelnamecode>0.1)

2.  (<computer><model>0.2)
    (<model><modelname>0.5)
    (<modelname><modelnamecode>0.2)
```

In this case using the product we get $T(x,y,z) = T(x,T(y,z)) = 0.009$ for the first path, while the second gets 0.02; again this estimate of importance that ranks path (2) above path (1) may not fully agree with users' intuition. The graph corresponding to our sample document, computed using the minimum as an aggregation operator is depicted in Figure 10. For the sake of clarity, only internal element nodes are shown.

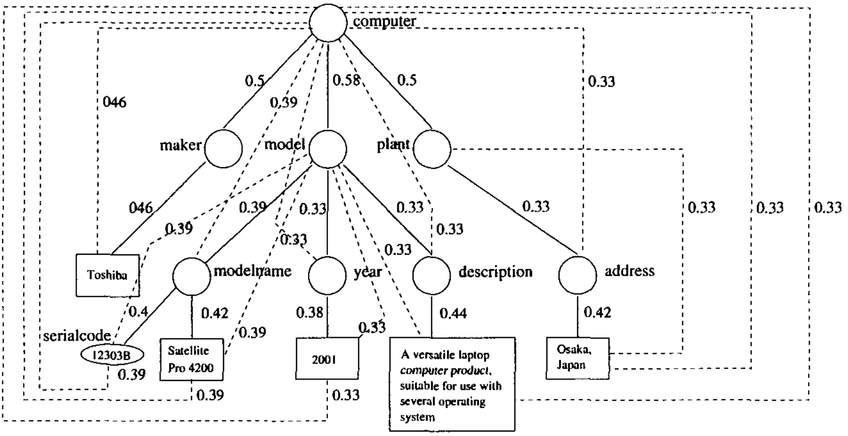


Figure 10. The closure of the XML tree corresponding to the document of Figure 1.

6 Query Execution

We are now ready to outline our query execution technique for well-formed XML documents, which relies on the following procedure:

1. Weight the target document tree and the query pattern according to the techniques described in Section 4.1. Weights on target documents can be computed once for all (in most cases, at the cost of a visit to the document tree). Though weighting the queries must be done on-line, their limited cardinality is likely to keep the computational load negligible in most cases.
2. Compute the closure of the target document's tree using a T-norm or a suitable fuzzy aggregation of the weights. This operation is dominated by matrix multiplication, and its complexity lies in between $O(n^2)$ and $O(n^3)$ where n is the cardinality of the node-set V of the target document graph. Again, graph closure can be precomputed once for all and cached for future requests.
3. Perform a *cut* operation on the closure graph using a threshold (this operation gives a new, tailored target graph). The cut op-

eration simply deletes the closure arcs whose weight is below a user-provided threshold α, and is linear in the cardinality of the edge-set.

4. Compute a fuzzy similarity matching between the subgraphs of the tailored document and the query pattern, according to selected type of matching. This operation coincides with the usual query execution procedure of pattern-based XML query languages, and its complexity can be exponential or polynomial w.r.t the cardinality of the node-set V of the target document graph (Comai *et al.* 1998), depending on the allowed topology for queries and documents (Cohen *et al.* 1993).

The first steps of the above procedure are reasonably fast (as document weights and closure can be pre-computed, required on-line operation consists in a sequence of one-step lookups) and does not depend on the formal definition of weights. The last step coincides with standard pattern matching in the query execution of XML query languages (Ceri *et al.* 1999), and its complexity clearly dominates the other steps.

Starting from the XQL query shown in Section 2.1, suppose to have the same simple query without the identification of the `<plant>` child node whose content is "`Japan`", i.e. suppose to have the following XQL query: `//[Computer contains "Japan"] (.//Model )` requesting all XML subtrees rooted in a `<computer>` node, whose content is "`Japan`". If we apply this query to the document of Figure 1, the query will return no answer because the `<computer>` node is not directely connected to any node whose content is "`Japan`", but if we apply the same query to the document of Figure 10 the query will return as output the `<model>` node, because the closure has introduced an edge between the `<computer>` node and the node whose content is "`Japan`".

7 A Logical Formulation

All graph-theoretical notions given in previous subsections can be readily translated in a simple logical formulation, to obtain an extensional fuzzy database.

First of all, we express the document graph as a conjunction of *ground facts*, e.g. instances of 1-ary and binary predicates *contains*, *value* and *content* with constant values. Typed predicates like *e-contains* and *a-contains* will be used to distinguish between element and attribute containment. For example, for the document in Figure 1 we have the following conjunction of facts:

e-contains(OID1-computer,OID2-maker) ∧
content(0ID2-maker, "Toshiba") ∧
e-contains(OID1-computer),OID3-model) ∧
a-contains(OID3-model,OID4-serialcode) ∧
value(OID4-serialcode, "1230B") ∧
...
e-contains(OID8-plant,OID9-address) ∧
content(OID9-address, "Osaka, Japan")

Then, we use the weighting procedure of Section 4.1 to estabilish importance, to be used as *truth-value* for the facts in the extensional database. For instance, using the fuzzy weighting model of Section 4.1, we have *e-contains(computer, maker)* = 0.5. Now we are ready to perform a *closure procedure* to augment the facts, according to the following *transitivity* rule:

$$e - contains(x,y) \Rightarrow le - contains(x,y) \quad (7)$$

$$le - contains(x,y) \wedge e - contains(y,z) \Rightarrow le - contains(x,z) \quad (8)$$

Formula 8 gives the truth-value of the new predicate *le-contains(x, z)* in terms of the truth-values of predicates *le-contains(x, y)* and *e-contains(y, z)*. Indeed, graph-based queries are inherently compound, raising the issue of finding the appropriate aggregation operator for combining the elementary truth-values. This is exactly the same problem of the choice of the t-norm to aggregate weights discussed in Section 4.1: selecting a conjunction means deciding if and how a lightweight intermediate element should affect the importance of a heavier element nested inside it. As we have seen, the straightforward approach to this problem is to use triangular norms, but the aggregation provided by t-norms may not coincide with users' intuition. However, our logical formulation allows us to see more clearly the association between conjunction and query execution semantics. Indeed, the conjunction to be employed can be a *logical*, *compensatory* or *product-based* one, depending on the user-selected semantics that was used to compute truth values of the initial predicates. In the following, we shall briefly discuss how the choice of a conjunction may affect query execution in our setting.

- **Logical conjunctions** are modeled by t-norms and express a conservative view in which the total degree of importance of a XML fragment is linked to the importance of its least important element. The most natural choice for conjunction, pure min, is the largest associative aggregation operator which extends ordinary conjunction. It is also the only idempotent one and, thanks to these properties, it well preserves query optimization properties (Fagin 1996). Once again, we note that using the min conjunction we adopt the most conservative attitude; unfortunately, as shown in Section 4.1, its behavior does not always coincide with users' intutition. An intermediate behavior is obtained by using $Lukasiewicz$ norm $T = max(a + b - 1, 0)$. Product-based conjunctions introduce a *probabilistic* view which also may create problems with user intuition (Section 5). They also pose other problems, as they are unfit for query optimization.
- **Weighted averages (WA)** promote a more *utilitaristic* view where the higher value of importance of an element can often *compensate* for a lower value of another one. In other words, it may happen that $WA(x, y) \geq min(x, y)$. Table 1 shows some

classical average-based choices for the aggregation operation. The degree of compensation for these operators depends on a tunable parameter $\gamma \in [0,1]$. We shall require this positive compensation to occur for all values of γ ; therefore we rule out operator A_1, which coincides with the min when $\gamma = 0$, and operator A_3, which coincides with the product. On the other hand, operator A_2 from Table 1 presents the single operand dependency problem, as it exhibits the absorption property (it always gives 0 when one of the operands is 0). Operator A_4 aggregates a conservative view with a utilitaristic one. When $\gamma = 0$, it coincides with simple arithmetic mean (operator A_5), which has been shown in previous examples and will be used in the sequel.

Table 1. Average-based fuzzy conjunctions.

norm	$T(x,y)$
A_1	$\gamma max(x,y) + (1-\gamma))min(x,y)$
A_2	$(x+y-xy)^\gamma (xy)^{1-\gamma}$
A_3	$\gamma(x+y-xy) + (1-\gamma)(xy)$
A_4	$\frac{\gamma min(x,y)+(1-\gamma)(x+y)}{2}$
A_5	$\frac{(x+y)}{2}$

The logical counterpart of the α-*cut* operation we performed on the weighed closure graph is thresholding truth values. Thresholding involves all predicates in the extensional database; intuitively, it will eliminate predicates having a low truth-value, providing a set of facts tailored to the user interests.

We now express the query as a logical formula, e.g.

$$Q = e - contains(x - plant, x - address) \wedge \\ value(x - address, \text{``}Osaka, Japan''') \qquad (9)$$

Q requires all plants whose address is Osaka, in Japan.

Note that *x-plant* and *x-address* are typed logical variables, where types are element or attribute names such as `car`, `maker`, `model`, `plant`. We shall write *x-t* to denote variable *x* belonging to type *t*. Matching the query formula to the transformed facts means to compute its truth-value, which is obtained by taking the conjunction of the truth-values of the atomic predicates. Consistently with our previous choice, the conjunction to be used is the same one that was used to compute the closure. Namely, we compute

$$\begin{aligned}\mu(Q) = \mu(e - contains(x - plant, x - address)) \wedge \\ \mu(content(x - address,'' Osaka, Japan''))\end{aligned} \qquad (10)$$

Once again we remark that the choice of the aggregation will affect query result; for instance, in the compensatory vision, there is no absorption property and $\mu(Q)$ may well be above zero even if either $\mu(e - contains(x - plant, x - address))$ or $\mu(content(x - address,'' Osaka, Japan'')$ are zero (but not both). More importantly, whatever the conjunction we use, the query result is a *ranked list* of couples $(x - plant, x - address)$, ordered according to their truth values.

8 Fuzzy Graph Matchings

In this section we shall briefly outline the fuzzy matching algorithm used for locating the fuzzy subgraphs of the extended document graph and computing their degree of matching with respect to the user query. To allow for maximum flexibility, several notions of matching can be employed. Here, we only outline their classification:

- *Lexical distance* Matching between document subgraphs and the query graph depends on the number of nodes they have in common, regardless of their position in the graphs. Different distance measures can be defined taking into account the fact that nodes may belong to different XML *lexical categories*, e.g. elements in the query graph may correspond to attributes in the document and viceversa.

- *Graph Simulation* Matching between document subgraphs and the query graph depends on the number of paths spanning the same nodes they have in common. Again, different distance measures can be defined taking into account the fact that nodes represent different types of XML lexical terms.
- *Graph Embedding* Matching between document subgraphs and the query graph is defined as a function φ associating query nodes to document nodes in such a way that edges and labels are preserved.
- *Graph Isomorphism* Matching between document subgraphs is a function φ as above, which in this case is required to be a one-to-one mapping.

We represent a graph, as usual, as a pair $Q = (N, A)$ where N is a set of nodes and A is a set of arcs.

The matching procedure consists of three steps:

1. Given the query $Q = (N, A)$, without taking membership values into account, locate a matching subgraph $G' = (N', A',)$ in the extended document graph (using, for instance, crisp depth first search), such that that there is a mapping $\varphi : V \to V'$ preserving arcs and arc labels[6]. A procedure `FindMatch` is used, according to the desired type of matching; in the case of graph embedding, its complexity is polynomial in $\mid N \mid$ for simple queries (Comai *et al.* 1998).
2. Compute the *ranking function* $J(Q, G')$ as follows:

$$J = \bigwedge_{(n_i,n_j)\in A} w_{arc}(\varphi(n_i), \varphi(n_j)) = T(w_{arc}(\varphi(n_i), \varphi(n_j), ...) \tag{11}$$

   In the second part of Eq. 11, we straightforwardly use t-norm associativity to compute the conjunction T over all arcs

[6] Moreover, this matching ensures that if values are specified on terminal nodes in the query graph, they also must appear as content labels of the corresponding nodes in the input document graph.

$\varphi(n_i), \varphi(n_j)$ in the document graph corresponding to arcs n_i, n_j in the query graph. This is the same procedure that was used for computing the truth-value of the sample query in Section 7. When T is the arithmetic average, we cannot rely on associativity and we get

$$\frac{1}{\mid A \mid} \sum_{(n_i,n_j) \in A} w_{arc}(\varphi(n_i), \varphi(n_j)) \tag{12}$$

Function 11 plays the same role as the objective function in standard fuzzy graph matching algorithms (Gold and Rangarajan 1996), expressing the degree of membership of a candidate subgraph in the result set as a conjunction of the weights on corresponding arcs.

3. Output the matching subgraph and its rank J.

9 Software Architecture

We shall now describe the architecture of ApproXML, a software system based on our approach. Before describing the architecture itself, we will shortly talk about the programming language chosen for its development.

9.1 A Short Introduction to Java and to the DOM API

Our software tool is implemented in the Java language using the DOM API (Application Protocol Interface).

Java is an advanced programming language developed by Sun, and has many advantages over other languages.

- The design of the language is such that it is extremely easy to learn; much easier than something like C++ (even though Java is just as powerful in many cases).

- Analogous to XML, Java is a language designed to be run (without any modification to the code) on virtually any platform. That is, the same programming instructions will work on a Macintosh computer, IBM PC, Sun workstation, etc. Unlike other programming languages, in which source code is directly compiled into machine specific language, Java compiles its programs into a format known as Java byte codes. Java byte codes can not be run without a program known as a Java "virtual machine." This virtual machine interprets the byte codes and then compiles the byte codes into machine specific languages. Thus when a Java program is transferred over the internet, what's really being transferred are the Java byte codes, which are then ready to be interpreted by the virtual machine present on the host computer.
- Java allows you to approach web projects with an unprecedented set of tools.
- It's free.

Programming can be viewed as composed by two aspects: data, and the instructions performed on that data (also known as procedures or methods).In traditional programming languages (C, Pascal, BASIC), there is a sharp separation between these two entities. Object Oriented languages, however, are designed such that you must group the data and their methods together very closely. While this may require a certain degree of extra thought at the 'design' stage of programming, the benefits are huge especially if you need to read and extend a previous software. The main terms and concepts that are characteristic of an object oriented language, and useful to follow the structure of this demonstrator tool, are:

- *Class* - a way to group data and methods together into one coherent package. When you create a class, you are defining the blueprint for an object.
- *Object* - a unique instance of a class. If a class is a blueprint for a building, think of an object as the actual building itself. Note that you can create many buildings out of the same blueprint, just as you can create many objects from the same class.

The **Document Object Model (DOM)** is a standardized object model for XML documents. DOM is a set of classes describing an abstract structure for an XML document. Programs that access document structures through the DOM interface can arbitrarily insert, delete, and rearrange the nodes of an XML document programmatically. DOM is being designed at several levels:

- Level 1. This concentrates on the actual core, HTML, and XML document models. It contains functionality for document navigation and manipulation.
- Level 2. Includes a "style sheet" object model, and thus defines functionality for manipulating the presentation information attached to a document. It also enables traversals on the document, defines an event model and provides support for XML namespaces.
- Level 3(still at a prototype level.) Will address document loading and saving, as well as content models (such as DTDs and schemas) with document validation support. In addition, it will also address document views and formatting, key events and event groups. First public working drafts are available.
- Further Levels (still under discussion). These may specify some interface with the possibly underlying window system, including some ways to prompt the user. They may also contain a query language interface, and address multithreading and synchronization, security, and repository.

The DOM API has been preferred to the SAX (Simply Application for XML) API, which although easy, and usually the most common for XML data, has a limited functionality that not allows to modifies or navigate the tree structure of the document.

9.2 Description of the Architecture

ApproXML is composed of two main modules, the `Pattern Locator` and the `Smusher`, corresponding to operations at two different levels of granularity. The `Pattern Locator` module is the core of our design. First, it parses and pre-processes the target document tree. Then, it uses a Match function to look for fragments of the target document having a *topological similarity* with the user pattern. The Smusher is a service module, which is called by the Match function of the Locator to perform XML *node smushing*, i.e. to evaluate similarity between elementary granules of information (such as XML nodes with their content and attributes) and create result nodes more suitable for user output. The final result of an execution is a list of smushed XML fragments, ordered according to their similarity with the pattern; this list is sent to a final Post-Processor module that organizes it in a catalog suitable for user consultation or further processing. Figure 11 depicts our architectural design.

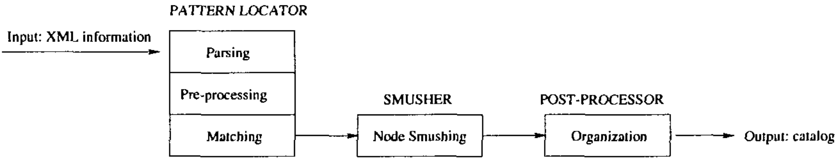


Figure 11. The architectural design.

The Pattern Locator's operation relies on the following procedure:

1. Parse the pattern, obtaining a standard DOM tree.
2. Parse the target input XML document, resulting in a weighted, extended DOM tree where nodes are ExtendedNode objects. Arc weights are computed once for all, at the cost of a visit to the document tree.
3. Compute the closure by visiting the extended DOM tree and calling the Closure method. The Conj function passed to Closure computes a fuzzy aggregation of the arc weights; choice of Conj is dataset dependent and can be done by taking user feedback into account (Damiani *et al.* 2000). Again, extended DOM closure can be pre-computed once for all and cached for future requests.

4. Perform a visit of the closure tree eliminating from each `Arc-set` the arcs whose weight does not reach a user-defined threshold (this operation gives a new, tailored extended DOM representation of the target document). Thresholding simply deletes the closure arcs whose weight lies below a user-provided threshold α, and costs an additional visit to the document tree.
5. Pass the pattern's and the target document's DOM trees to a `Match` function to evaluate the *similarity matching* between the subtrees of the tailored target document and the pattern tree. In turn, `Match` uses the services of the `Smusher` module to evaluate similarity between nodes.
6. All the results are provided to the user in a list ordered from the most relevant (highest matching value) to the less. The user see a list of ordered XML document fragments.

This tool was tested and developed on a Sun Unix System (Solaris) and, at the same time, on a Windows pc. For this work we used Java 1.2 including packages `java.awt` and `javax.swing` for the graphical interface utilities. Using a Sun System, we decided to use the Sun parser (shareware), which was used for the little editor (shareware) we extended for this project. Start point of this tool was `DomView 1.0` author Sun Koh at The BeanFactory, a little XML editor which displays contents of an XML DOM object and shows the DOM tree in a graphical way. DomView is shareware and it can be used and modified under the terms of the GNU General Public Licence.

9.3 Description of the Menu Items

- *File:* this menu allows to open an XML file (Open),or a weighted XML file (OpenWeighted). To open a file (weighted or not), you need to insert the complete path in the appropriate window. The opened file is shown in the left side of the main window. Now it is possible to insert the weights (if the document is not already weighted), following these steps:

  - completely expand the document tree clicking on all the symbols,
  - then for each node, chosen in the right order (descending from the root), insert the arc weights.

  See parts 1 and 2 of the architecture description.
- *Bi-Closure:* this menu allows to compute the bi-directional transitive closure of the weighted document tree(i.e. the transitive closure of the document seen as an undirected graph). At the moment only the *min()* function is implemented, but the tool could be easily expanded with other functions.
- *Oriented-Closure:* this menu allows to compute the transitive closure of the weighted document tree, this time seen as a directed graph). At the moment only the *min()* t-norm is implemented, but the tool could be easily expanded with other t-norms. See part 2 of the architecture description.
- *Threshold:* this menu opens the window which allows to insert a threshold value to prune the closed tree deleting all arcs with a value lower than the inserted threshold (InsertThreshold). This functionality is described in part 4 of the architecture.
- *Query:* this menu allows to create and save a new query writing a well-formed XML fragment (Pattern) in an appropriate window (CreateQuery), or to open a saved query indicating its complete path (OpenQuery).
- *Extract:* this menu allows to extract the matching result of a single document (Extract) or to see the complete ranked list of the results obtained from a data-set (List). See parts 5 and 6 of the architecture description.

9.4 Description of the Most Important Classes

We enumerate in the sequel the main classes of our java code referring to the architecture described:

a) *FlexSearch* is the main frame, this class starts the software, sets the main panel and contains the function that closes the tool. It also calls all the other classes.

b) *DomViewPanel* is the most important class of the application and uses all the following ones. It:

- displays the tree implementations of the XML document
- displays the tree implementations of the XML query
- displays the attributes of a single node of the XML document
- contains a variety of functions that gives information or implements:
  1. document representation
  2. the matching query-document
  3. document opening, document weighting, query opening

c) *DomViewUtils* contains the calls to the Sun parser. It has two methods `getSunDocument(String location)` that, given an XML file location (a complete path), returns a document object and, viceversa, `setSunDocument(String text)` that saves a document object (given in a string format) as an XML file. Both methods return a document object. See part 1 of the architecture.

d) *OpenFileDialog* creates the window where it is possible to write the complete path of the XML file to open. The text of the location, contained in a JTextField, is passed to the `getSunDocument(String location)` described above. It contains two buttons (Jbutton) - Open and Cancel - activated respectively by the `protected inner class OpenAction`, that calls the method `openFile` defined in the class DomViewPanel, and by the `protected inner class CancelAction` that, simply, close this dialog window.

e) *WegInput* offers two different possibilities :to insert the arc weights manually in the DOM tree or to compute them automatically according to the techniques described in Section 1.4.1 . All weights (stored as float numbers) are inserted into a vector - branch_Weigth - static because it has to be given to the closure part of the software.

f) *OpenWeigthFileDialog* opens in the left part of the main window an XML file that was already saved with all its weights. The most important function is `OpenWeightAction` which calls the method `openWeightFile( String text)`, defined in the class DomViewPanel, that get the DOM of the XML file called, reading the weights contained in the apposite comments and filling up the weights vectors. These classes implement the part 2 of the architecture.

g) *NewQueryDialog* creates the window that allows to write the pattern of a new query with the only restriction that it must be a well-formed XML fragment.

h) *OpenQueryDialog* is a class analogous to OpenFileDialog, created for the query tree. It opens a window that get the location (complete path) of an XML file containing a query pattern.

i) *AdjacencyMatrix* is the class that represents the adjacency matrix of the document DOM tree. The method `float[] [] fillUpArcsMatrix(Vector v, float[] [] m)` creates a bi-directional closure, based on the function *min()*, matrix of the document tree getting as input the float weights vector and the matrix to fill up, and giving the filled matrix as output.

l) *QueryMatrix* contains the adjacency matrix of the pattern tree. These classes implement the part 3 of our architecture.

m) *Threshold* opens a window to get the threshold value (float)and prune the closure tree of the document. See part 4 of the architecture.

n) *Matching* contains all methods for extracting the matching document fragment, using the sub-matrix searching method. The output of this class is a new adjacency matrix containing only those node and arcs of the document that match the query pattern. See part 5 of the description if the architecture.

o) *Result* is a complex structure containing in a single object the extracted result fragment, its matching value, computed according to the techniques described in Sect. 1.8, and the path of the original document. It is useful to prepare the vector containing all the results extracted from a data-set. The ordered vector is used to print all results in a list, starting from the highest matching value fragment to the lowest.

p) *VisOutput* contains the window in which the matching fragment of the document in exam is showed. The most important method of this class transforms the result matrix in an XML well-formed fragment, deleting from each extracted document node all useless children nodes leaving only those requested by the query.

q) *VisResult* contains the window in which all the fragments, extracted from a data-set, are shown in order from the highest matching one to the lowest. See the last part of the architecture.

10 Internet-Based Applications

Throughout the chapter we have described a flexible technique for searching patterns in XML datasets. Besides searching, evaluating the similarity between a user pattern and XML data has a number of potential applications. In this section, we briefly mention some of the application fields where flexible XML searching can be used to provide new Internet-based products and services or for the enhancement of existing ones.

- *Brokerage Applications:* An increasing wealth of information (e.g. about available WWW services) is made available in XML format via the Web or in special purpose repositories. Brokerage applications need to select the information more closely related (though, perhaps, not exactly corresponding) to the user interests and to present it in a customized format suitable for human- and machine reading and for reporting applications.
- *Foreign Documents Management:* In XML-based *e-business* systems, huge amounts of XML information cross organizational borders, where document structure may change while retaining commonalities in tag repertoire and vocabulary. Approximate techniques for location and restructuring of XML information granules in foreign documents look promising for improving the quality of e-business information interchange.

- *Web-mining systems and Vertical Portals:* The term "Web mining system" indicates a wide class of applications/services collecting domain or market-place specific information from the Web. Such information is organized and presented to users in order to allow them to keep under control the information flow about their field of interest, possibly applying data-mining techniques. Current systems use Web spiders for data-collection and keyword-based clustering to organize and query HTML data. The availability of such an information flow in XML format is an occasion to provide high quality Web mining services via advanced reorganization and search tools specific for XML.

References

Bardossy, A., Duckstein, L., and Bogardi, I. (1993), "Combination of fuzzy numbers representing expert opinions," *Fuzzy Sets and Systems*, vol. 57, no. 2, pp. 379-389.

Bordogna, G., Lucarella, D., and Pasi, G. (1994), "A fuzzy object oriented data model," *IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 313-317.

Bosc, P. (1998), "On the primitivity of the division of fuzzy relations," *Soft Computing*, vol. 2, no. 2.

Bouchon-Meunier, B., Rifqi, M., and Bothorel, S. (1996), "Towards general measures of comparison of objects," *Fuzzy Sets and Systems*, vol. 84.

Box, D. (2001), *XML*, Developmentor Series, Addison-Wesley.

Ceri, S., Comai, S., Damiani, E., Fraternali, P., Paraboschi, S., and Tanca, L. (1999), "XML-GL: a graphical language for querying and restructuring XML documents," *Computer Networks*, vol. 31, pp. 1171-1187.

Ceri, S. and Bonifati, A. (2000), "Comparison of XML query languages," *SIGMOD Record*, vol. 29, no. 1.

Clarke, C.L.A., Cormack, G.V., and Burkowski, F.J. (1995), "An algebra for structured text search and a framework for its implementation," *The Computer Journal*, vol. 38, no. 1.

Cluet, S., Deutsch, A., Florescu, D., Levy, A., Maier, D., McHugh, J., Robie, J., Suciu, D., and Widom, J. (1999a), "XML query languages: experiences and exemplars." Available online at http://www-db.research.bell-labs.com/user/simeon/xquery .html.

Cluet, S., Jacqmin, S., and Simeon, J. (1999b), "The new YATL : design and specifications," Technical Report, INRIA.

Cohen, R., Di Battista, G., Kanevsky, A., and Tamassia, R. (1993), "Reinventing the wheel: an optimal data structure for connectivity queries," *Proc. of ACM-TOC Symp. on the Theory of Computing*, S.Diego, CA, USA.

Cohen, S., Kogan, Y., Nutt, W., Sagiv, Y., and Serebrenik, A. (2000), "EquiX: easy querying in XML databases," *WebDB* (Informal Proceedings).

Comai, S., Damiani, E., Posenato, R., and Tanca, L. (1998), "A schema-based approach to modeling and querying WWW data," in Cristiansen, H. (ed.), *Proceedings of Flexible Query Answering Systems (FQAS '98)*, Roskilde (Denmark), Lecture Notes in Artificial Intelligence 1495, Springer.

Damiani, E. and Tanca, L. (1998), "Semantic approaches to structuring and querying Web sites," in Spaccapietra, S. and Maryanski, F. (eds.), *Data Mining and Reverse Engineering (Proceedings of the International IFIP Working Conference on Database Semantics DS-7)*, Zurich, Switzerland, Chapman & Hall, Jan.

Damiani, E. and Tanca, L. (2000), "Blind queries to XML data," *Proceedings of DEXA 2000*, London, UK, September 4-8, Lecture Notes in Computer Science, vol. 1873, Springer, pp. 345-356.

Damiani, E., Tanca, L., and Arcelli, F. (2000), "Fuzzy XML queries via context-based choice of aggregations," *Kybernetika*, vol. 16, no. 4.

Damiani, E., Oliboni, B., and Tanca, L. (2001), "Fuzzy techniques for XML data smushing," *International Conference, 7th Fuzzy Days*, Dortmund, Germany, October 1-3, Lecture Notes in Computer Science, vol. 2206, Springer, pp. 637-652.

Deutsch, A., Fernandez, M. Florescu, D., Levy, A., and Suciu, D. (1999), "A query language for XML," *Proceedings of the WWW-8 Intl. Conference*, Canada.

Dubois, D. and Prade, H. (1996), "What are fuzzy rules and how to use them," *Fuzzy Sets and Systems* 84.

Dubois, D., Esteva, F., Garcia, P., Godo, L., Lopez de Mantaras, R., and Prade, H. (1998), "Fuzzy set modelling in case-based reasoning," *Int. Jour. of Intelligent Systems*, vol. 13, no. 1.

Dubois, D., Prade, H., and Sedes, F. (1999), "Fuzzy logic techniques in multimedia database querying: a preliminary investigations of the potentials," in Meersman, R., Tari, Z., and Stevens, S. (eds.), *Database Semantics: Semantic Issues in Multimedia Systems*, Kluwer Academic Publisher.

Fagin, R. (1996), "Combining fuzzy informationm from multiple systems," *PODS'96*, Montreal, Canada, June, pp. 216-226.

Gold, S. and Rangarajan, A. (1996), "A graduated assignment algorithm for graph matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2.

Klir, G. and Folger, J. (1988), *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall.

Radecki, T. (1979), “Fuzzy set theoretical approach to document retrieval,” *Information Processing and Management*, vol. 15, pp. 247-259.

Robie, J. (1999), “The design of XQL.” Available online at http://www.texcel.no/whitepapers/xql-design.html.

Robie, J., Chamberlin, D., and Florescu, D. (2000), “Quilt: an XML query language.” Available online at http://www.almaden.ibm.com/cs/people/chamberlin/usecases.html.

W3C (1998a), “Extensible Markup Language (XML) 1.0,” Feb. Available online at http://www.w3c.org/TR/REC-xml/.

W3C (1998b), “Document Object Model (DOM),” W3C Recommendation 1 October. Available online at http://www.w3.org/DOM/.

W3C (1998c), “QL’98 - The Query Languages Workshop.” Available online at http://www.w3.org/TandS/QL/QL98/.

W3C (1999a), “Resource Description Framework (RDF) Model and Syntax Specification,” W3C Recommendation 22 February. Available online at http://www.w3.org/TR/REC-rdf-syntax/.

W3C (1999b), “XSL Transformations (XSLT) Version 1.0,” W3C Recommendation 16 November. Available online at http://www.w3.org/TR/xslt.

W3C (2000), “Extensible Stylesheet Language (XSL),” Version 1.0, October. Available online at http://www.w3c.org/TR/xsl/.

World Wide Web Consortium (2001), “XQuery 1.0: an XML Query Language,” W3C Working Draft 07 June. Available online at http://www.w3.org/TR/xquery/.

Chapter 4

Agent-Based Hypermedia Models

W. Balzano, P. Ciancarini, A. Dattolo, and F. Vitali

Hypermedia models have been conceived to capture and describe the fundamental characteristics of hypermedia systems and of their implemented and desirable features. From the Dexter model onward, many have worked on theoretical analysis of hypermedia systems, including the most famous of such systems, the World Wide Web.

Agents, considered as software modules showing autonomy and persistency (or abstractions with well-defined state and behavior) have been a reasonable, although not universal, tool in the description of hypermedia systems.

The concept of agent allows complex behavior to be modeled efficiently and in a meaningful manner. This is even more true for agent-based hypermedia systems, where the concept of agents is explicitly used in the implementation of the system itself.

1 Introduction

The current availability of inexpensive network technologies has revolutionized the traditional perspective of hardware (see for instance network computers) and software (agent-based software). Early collaborative hypermedia systems are now referred to as "closed" architectures due to their reliance upon monolithic design, their inability to support integration and distribution of data and tasks and their lack of meaningful communication protocols.

Advances in research have given rise to a new generation of "open" collaborative hypermedia systems characterized by modular architectures and strong utilization of distribution and communication.

From a software standpoint, there is an interest in viewing software as a collection of agents that interact by coordinating knowledge-based processes (Bieber *et al.* 1997, Gasser 1991). In this way, software can be conceived as an open system, "a large-scale information system that is always subject to unanticipated outcomes in its operation and new information from its environment" (Hewitt 1991). As in almost every field of computing, agents can be used to support many aspects of information systems, including search and retrieval, personalization, customization, maintenance and classification of content.

The story of software agents begins with the idea of a 'soft robot' - semi-autonomous programmed entities capable of carrying out tasks toward a goal, while requesting and receiving advice and orders in human terms. In recent years, a much narrower marketing-oriented use of the term "agent" has emerged, with a fairly tenuous relationship to actual agent technologies and a steady growth despite its disappointing failures. This has lead to a partial transformation of the concept of agent into an anthropomorphized, self-customizing virtual servant designed for a single task: a pleasing interface to a world of information that does not please us.

Agents (Maes 1994, Wooldridge and Jennings 1995) are abstractions which encapsulate state and behavior. The main difference between an agent and an object is that an agent is autonomous, has full control on its behavior and can invoke services form the outside world, while objects are usually conceived as reactive abstractions. Agent-based systems are described by their cooperation model (i.e., what kind of society – collaborative, authoritative, etc.), implemented by the agents, their coordination model (i.e., how, how often and with which roles is the dialogue among agents performed) and their composition model (i.e., the technologies used for letting the agents execute, communicate, move, interoperate).

Hypermedia models, whose most famous example is the Dexter Reference Model (Halasz and Schwartz 1994), have tried to precipitate the basic characteristics of hypermedia systems both available and conceivable. Not surprisingly, some recent advancements, and in particular the addition of the concept of autonomous modules providing system-wide functionalities, do not easily fit within these older systems. Agent-based hypermedia requires substantial modification in traditional hypermedia models.

We should also consider that, although the WWW (World Wide Web) (Berners-Lee *et al.* 1994) can be positioned fairly precisely in the taxonomy of hypertext systems, its wide availability and simple architecture has made it the standard environment for such advancements, so that it is important to acknowledge the specific assumptions and characteristics of the Web when discussing agent-based hypermedia.

This chapter is structured as follows: in Section 2 we discuss hypermedia from a principled perspective, trying to summarize the basic concepts of the field and trace the evolution of hypermedia models. In Section 3 we introduce some agent-based hypermedia models, discussing similarities and differences. The set of hypermedia models discussed in this chapter cover important issues in hypermedia field, such as adaptive and multimedia aspects, link service, CSCW (Computer Supported Cooperative Work), etc. Section 4 ends the chapter.

2 Hypermedia Models

The aim of this section is to introduce a definition of hypermedia system and trace the evolution of hypermedia models from the first reference model, Dexter, towards current open hypermedia approaches. We describe in detail the Dexter model, because we believe that the knowledge of its basic organization is important for a correct understanding of the subsequent agent-based hypermedia models.

2.1 Definition of Hypermedia

The human mind does not operate in a strictly linear manner. Our succession of thoughts tend to form associations – when we think of something, we will also think of something else that is related to it. We thus jump quickly from one topic to another related piece of information. The *hypertext* represents a attempt to model this nonlinear association with information repositories. Self-contained pieces of information are linked together by natural or topical association rather than organizing them in the familiar paper-based book sequential structure.

Hypertext is a metaphor for organizing textual information in a complex, non–sequential web of associations that allows the user to browse through related topics, regardless of the presented order of the topics. The term was first used by Ted Nelson (1987) in describing his Xanadu system.

The word "hypertext" suggests that all information is in the form of plain text. *Hypermedia* is an acronym which combines the words "*hyper*text" and "multi*media*" to refer to a hypertextually connected combination of multiple media, such as text, sound, and/or motion video. Thus, hypermedia is an augmented (or generalized) hypertext because it incorporates multimedia, enabling the user to selectively navigate through not only text, but virtually any kind of information that can be stored electronically. In the literature the terms of hypertext and hypermedia are often used interchangeably and we will make the same assumption here.

A *hypermedia* in general is modeled as a network of nodes that are connected through a set of links. The node usually represents a concept and the links connect related concepts. The nodes can be either atomic or composite; the links can be defined among any set of multimedia objects, including sound, motion video, and virtual reality.

Together nodes and links form what we call a hyperdocument; it can be viewed as a graph, which may be arbitrarily complex. In Figure 1 we show a simplified view of an small hyperdocument, having only five nodes (A, B, C, D, E) and six links. Figure 1 also shows that links are tied to specific points (or words or regions) within a node, called *anchors*.

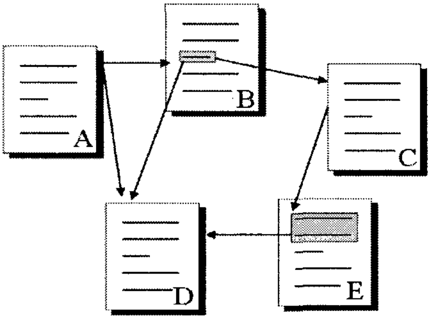


Figure 1. A simple example of hyperdocument.

2.2 Dexter and Beyond

The number of available hypermedia systems, based on and described by different formalisms and features and of varying levels of sophistication and usability, prompted a few years ago the design of a common reference model, in order to create an agreed-upon terminology and a conceptual framework "*to unify the disparate notions of node and link, to augment link-based networks with other structures, to integrate hypermedia with existing environments and content editors, represent and enforce different data models and runtime behavior, store properties of node presentations, interchange hypertext across systems*" (Gronbæk and Trigg 1999).

An important result in this direction is the Dexter model (Halasz and Schwartz 1994); it is an attempt to capture, both formally and informally, the important abstractions found in a wide range of existing and future hypertext systems. The goal of the model is to provide a principled basis for comparing systems as well as for developing interchange and interoperability standards. The model is divided in three layers, with glue in between, as shown in Figure 2.

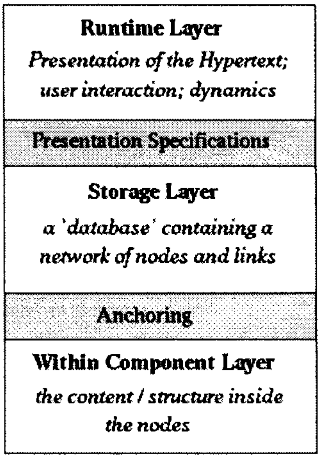

Runtime Layer
Presentation of the Hypertext; user interaction; dynamics
Presentation Specifications
Storage Layer
a 'database' containing a network of nodes and links
Anchoring
Within Component Layer
the content / structure inside the nodes

Figure 2. The Dexter model.

The *run-time layer* provides a model for the presentation mechanisms, i.e., what is made available to the user to access, view and manipulate the hypermedia network structure; it is intended to cover the dynamic aspects of hypermedia systems.

The *storage layer* is where the focus of the model is concentrated; it models the basic node/link network structure that is the essence of hypertext; it describes a "database" that is composed of a hierarchy of data-containing "components" (normally called "nodes") which are interconnected by "links". The storage layer focuses on the mechanisms by which the components and links are "glued together" to form hypertext networks. The components are treated in this layer as generic containers of data.

The *within component* layer is concerned with the contents and structure of the information *within* the components of the hypertext network. This layer is purposefully not elaborated within the Dexter model, since it is dependent on the type of media, the data format, etc.

Between the run-time layer and the storage layer Dexter introduces an interface responsible for managing *presentation specifications*, i.e., instructions on how a component should be presented to the user.

Between the storage layer and the within component layer Dexter specifies another interface that provides the basic navigation mechanism, allowing users to access components through anchors. The *anchoring* mechanism specified by the Dexter model allows span-to-span links to be supported while maintaining a separation between the two layers.

Other reference models have been developed: some extend Dexter, such as the Amsterdam Hypermedia Model (Hardman *et al.* 1994), while others are based on different information structures (see for instance Trellis (Stotts and Furuta 1989), based on Petri nets), or formalisms, such as the formal model of Lange (1990), formally defined in the VDM specification language.

The evolution of hypermedia models and systems can be captured from the taxonomy proposed by Osterbye and Wiil (1996); it classifies existing hypermedia systems into five categories:

- *Monolithic models.* The systems modeled by this approach are characterized by having one module which is responsible for all the aspects of the systems. Examples of monolithic systems are KMS (Akscyn 1988) and NoteCards (Halasz 1988).
- *Hyperbase models.* The systems modeled by this approach are characterized by a storage layer which manages both contents and structure and a session manager that assists viewers in main-

taining contents and structure of the hypermedia. Examples of hyperbase systems are Neptune (Delisle and Schwartz 1986), Sepia (Streitz *et al.* 1992) and EHTS (Wiil 1992).

- *Embedded link models.* The systems modeled by this approach are characterized by two layers, a storage and a runtime layers. They are special cases of hyperbase systems although they do not explicitly separate contents from structure. An important example of the embedded link model is the WWW (Berners-Lee *et al.* 1994).
- *Link server models.* The systems modeled by this approach are characterized by using third-party viewers to present and store the contents part of the hypermedia. In these systems the links and other hypermedia structures are separated from the data. A link servers contains a linkbase and a session manager is responsible for assisting third-party viewers in maintaining structure. Examples of link server systems are Sun's Link Service (Pearl 1989), Microcosm (Hall *et al.* 1996) and Multicard (Rizk and Sauter 1992).
- *Open hyperbase models.* This approach combines the hyperbase and link server approaches: the systems modeled by this approach are characterized by a storage module (responsible for storing both the hypermedia structure and the contents) and a session manager (responsible for assisting third-party viewers in maintaining structure and storing contents). Examples of open hyperbase systems are DHM (Gronbæk and Trigg 1994), HyperDisco (Wiil and Legget 1996), HyDe (Dattolo and Loia 2000).

3 Agent-Based Hypermedia Models

Agents (Wooldridge and Jennings 1995) are abstractions which encapsulate state and behavior. Both agents and objects encapsulate state. Objects encapsulate some form of control, in the form of methods. Agents encapsulate both control and an ontology, that is, a vision of the world of interest for the agent itself. In fact, the main difference between an agent and an object is that an agent is autonomous, i.e., it is using its ontology, has full control on its behavior and can invoke services form the outside world. Instead, objects are usually conceived as reactive abstractions, and, more importantly, they are more granular, since they have been invented to be easily reused.

Even more striking is the difference between a system made of objects and a multiagent system (Ferber 1999). For instance, the static structure of the code of an object-oriented system is usually described by a class diagram. The attempt to describe in a similar way a society of agents would be quite useless, because concepts like the beliefs of an agent, its ontology-driven proactive behavior, or its "social abilities" are simply not captured by a class diagram. In fact, systems composed of agents require specific forms of specification and design methods.

There are at least three abstraction levels that need to be dealt with in an agent-oriented environment:

- *the cooperation model*: systems made of agents resembles societies; at this abstraction level it is important to analyze and specify both the roles that an agent has to play and the social protocols that an agent has to support in order to interact with other agents. It is at this level that a society of agents is initially described, usually specifying which roles agents can play, i.e., which behaviors they will enact and which laws will rule their interactions. For instance, at this level we could describe a society of agents as *tyrannical*, or *strongly centralized*, because there exists an agent with the power to dictate and monitor the behavior of other agents. This kind of decision would have a strong impact on the nature of the *tyrant agent*, because it would probably assign to it special responsibilities not required for the other *subject agents*. On the contrary, in a society of agents based on a *contract net*, we would find agents playing sometimes the role of *contract givers* and sometimes of *contract tak-*

ers. We could also specify a society by combining simpler cooperation models: for instance, a *tyrannical contract net* is a society where all contracts are subject to approval by a single agent, namely the tyrant.

- *the coordination model*: while the cooperation model is fully abstract, and related only to the forms of interaction which have to be supported, the coordination model introduces some architectural choices. For instance, at this level an agent could be classified as a client, or a server, and consequently it can be designed accordingly. The coordination model is probably independent and orthogonal to the cooperation model. For instance, in a tyrannical society, we could make the tyrant a server, and the other agents clients, or, vice-versa, the tyrant could be a client, and other agents could be a federation of servers.
- *the composition model*: this level is even more implementation-oriented than the coordination model, and includes decisions on which technologies should be used to actually build an agent. For instance, at this level an engineer should decide which software component technologies can be used, like for instance JavaBeans or XML-based modules. This level is also quite orthogonal to both the cooperation level and the coordination model.

In the field of hypermedia, agents are heavily used in the support of hypermedia functionalities, (such as system customization, information retrieval, or collaboration support). On the other hand, only a few implementations exist whose formal model is based on agents. In the following subsections we show a few of them.

3.1 HyDe (Hypermedia Distributed Design)

HyDe (Dattolo and Loia 2000) is a concurrent distributed hypermedia model based on an extension of the actor model (Agha 1986).

Briefly, the actor model can be presented as a universe containing several computational agents, called actors; actors are a class of computational agents which carry out their actions in response to incoming communications; they encapsulate procedural and declarative information into a single entity. Actors perform computation through asynchronous, point-to-point message passing; each actor is defined by its state, a mail queue to store external messages and an internal behavior; an actor reacts to the external environment by executing its procedural skills, called scripts.

In HyDe there is no monolithic resource responsible for the global management of the hypermedia, but an aggregation of autonomous and independent actors, each of which owning a behavioral responsibility and a partial perception of the other members of the actor community.

Each node in HyDe corresponds to an actor. An actor allows passive information in its *acquaintances*, which are slots containing data. On receiving a stimulus, the actor may modify its internal status and interact with the external environment; these actions are performed by scripts, which are local functions associated with the actor. The social activity of an actor, i.e., the ability to establish collaborative goals, is possible because of the ability to contact "neighbor" entities.

Figure 3 shows the complete architecture of HyDe, based on the two-layer abstraction proposed by Dexter: storage and run-time layers.

The *storage layer* constitutes the structure of the hypermedia as provided by its author. The main purpose of this layer is to maintain the persistent objects, the collection of which defines the hypermedia in terms of dynamic internal mechanisms. This layer is organized in the *Structural* level, that contains atomic nodes (HypActors) and links (HypLinks), and the *Meta* level, that is constituted by composites (named Collectors).

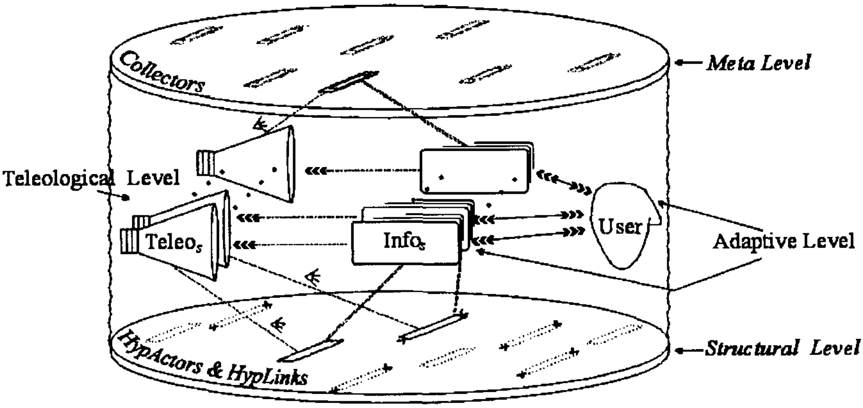


Figure 3. Architecture of HyDe.

The *run-time layer* represents the part of HyDe devoted to support the (adaptive) presentation of the storage components to the user. This layer is organized in the *Teleological* level, that provides all the possible dynamic user perspectives of the node and allows the used to interact with the data and services provided by a certain HypActor (Collector or HypLink), and the *Adaptive* level, that monitors the users' behavior and records their actions on the hypermedia nodes.

HyDe modifies the way in which complex software systems, such as hypermedia, are generally conceived. The main difference compared to other significant proposals (Garzotto *et al.* 1995, Gronbæk and Trigg 1994, Stotts and Furuta 1989) is in the *distribution* of not only data, but also of *control*, and in the fundamental role of *communication*. As a result, the storage and run-time layers are composed of *active* entities, that embody enough knowledge to solve global goals by cooperative activities.

3.2 CoHyDe (Collaborative HyDe)

CoHyDe (Balzano *et al.* 2000) represents an open, distributed collaborative extension of HyDe toward collaboration.

Geographically distributed project teams require a better support for full interaction. An open collaborative framework should favor activities performed by geographically and temporally distributed groups, supporting (a)synchronous work, notification of events and awareness tools.

As mentioned, a distributed group support system needs to address four important issues to allow distributed collaborative work: distribution, communication, coordination and cooperation. CoHyDe embodies these four aspects as the main architectural features of the model.

CoHyDe supports temporally and geographically distributed workgroups. A feature of the model is the extensive reference to collaboration within two logical levels: the usual user level and the architectural level. In fact, in order to achieve common objectives, both the members of a workgroup and the actors that constitute the internal framework cooperate using the communication protocols and tools.

Figure 4 shows the CoHyDe architecture, designed for collaboration activities: it is organized in three layers: Coordination, Access and Work layers. Each layer performs activities of distribution, communication, coordination and cooperation during the interaction with the other layers.

- The *Coordination Layer.* The (a)synchronous coordination of the collaboration activities is the aim of the Coordination layer. This layer is composed of Collaboration actors, C for short. A crucial point for successful collaborations is the manner in which individual work is related to the group as a whole. Co-workers, under changing and unpredictable conditions, make autonomous decisions when working alone which the group cannot foresee or plan. To enable a separated group of co-workers to collaborate, they need to coordinate themselves (Malone and Crowston 1994). Uncoordinated interactions often suffer from serious problems, such as lack of focus, lack of convergence (Romano *et al.* 1997), presence of redundant work and undue time to completion of the work (Schlichter *et al.* 1997). The coordination activity of the Coordination layer is made more effective since it is supported by intensive cooperation, based on continuous communication flows that the C establish with the actors in the other two layers. Each participant to a collaboration can create a cooperative session. All the sessions created within a collaboration are managed and coordinated from the same agent and from it they inherit data and functionality. C is a composite entity and can be viewed as the organization of internal data/scripts and of a collection of sessions that evolve in the time. Each session is an actor that inherits from the agent class a subset of tasks, a subset of users and specific constraints and abilities.

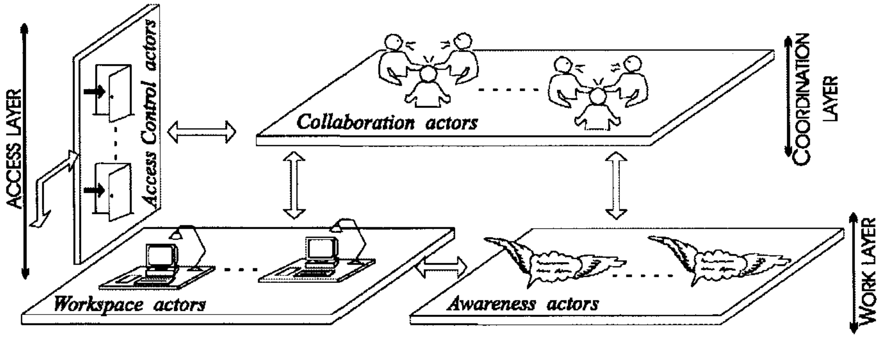


Figure 4. Architecture of CoHyDe.

- The *Access* layer restricts access to collaborations and to information. This layer is composed of Access Control actors, or AC for short. There is a AC for a each user. The AC is responsible of the parameters of the collaboration activities, of the maintenance and update of the roles and the access rights of the co-workers. The roles, duties and access rights on documents of a single user depend on the different collaborations and often change during the same collaboration according to the evolution of the tasks. In order to manage this knowledge dynamically, the AC maintains an active communication with the Coordination layer and a continuous cooperation with the user workspace. The communication with the user workspace is very intensive since any writing or modification operation of the user needs appropriate verifications.

- The *Work* layer enables the distribution of data and tasks, augmenting the awareness of the users and their cooperation. It manages the workspaces of the users, by following their working activities and performing the (a)synchronous process of notification. The Work layer is composed of two populations of actors: Workspace actor (W), and Awareness actor (A).
  - *Workspace* level. Each W actor manages the interface between the system and the user, allowing him/her to perform private and shared activities through virtual interfaces. Virtual workspaces (Romano *et al.* 1997) allow work to be accomplished independently of any one's specific time constraints; besides, they provide for simultaneous interaction of local and remote teams as well as rapid acquisition of feedback on material that must be reviewed by the whole group.
  - *Awareness* level. This level aims to make the user aware of the collaboration process. Collaboration cannot be separated from the concept of group awareness. CoHyDe supports some consolidated types of awareness (Sohlenkamp *et al.* 1997), such as organizational awareness (the knowledge of how the work group fits in with the larger purposes of a project or society), structural awareness (the roles, tasks and purposes of the people involved in the collaboration), work-

space awareness (up-to-the minute knowledge about the state of another's interaction with the workspace), domain awareness (the information and tools that are specific to the application domain, and helps the user to better understand the actions and choices of the other co-workers.)

3.3 Microcosm

Microcosm (Hall *et al.* 1996) is an implementation of the modern concept of Open Hypermedia System (OHS). An OHS is more than simply a hypermedia system relying on an open protocol: on the contrary, open hypermedia architectures store and manage information about links between multimedia documents separately from the documents themselves. Links are "first class" objects, and a link service applies the links to the system's data.

Web documents contain links in HTML; in this sense, the WWW remains a "closed" hypermedia system. However, the basic infrastructure of the WWW does not inherently prevent abstracting links from documents and maintaining them separately. Microcosm models the hypermedia system as a collection of loosely coupled processes (agents) running in parallel, and communicating by messages.

The linkbase describes a set of links; a link service may operate with multiple linkbases. The DLS (Distributed Link Service) (Carr *et al.* 1995) provides a link management and delivery service for WWW documents. In the same way that a client connects to a remote Web server to access a document, DLS allows the client to connect to a link server to request a set of links to apply to the data in a document. Buttons and explicit anchors are rejected as primary navigation mechanism; rather, links should be applied to all kinds of application-specific document types, either via a publication-time compilation of the links into the application's proprietary hypermedia link format, or via browse-time user queries on keywords, phrases or sections.

The architecture of the DLS is based on agents, which have proven to be an appropriate paradigm for engineering the link service. The main agents in this model are:

- the *Link Resolution Agent* (LRA), which are used to interrogate the linkbases. This architecture promotes scalability, by allowing multiple LRAs and replication of linkbases across the network, The LRAs may be aided by other agents for link generation and maintenance. For example, an agent may extract links from a set of HTML documents in order to build a linkbase; another agent may check link integrity to avoid "dangling pointers".
- The *Distributed Link Server* (DLS) agent itself, which inserts links into the documents on the fly before sending them to the requesting browser,
- The *Distributed Information Management* (DIM) agent, that provide resource maintenance and integration. Open hypermedia and agent paradigm provide a powerful approach to distributed information management. An example of DIM agents is the integrity agent of Microcosm, that deals only with the content of linkbases and does not need to access the documents' content. DIM agents are autonomous, sometimes reactive, have social ability and when appropriate they can be mobile.

Within the Microcosm team, some further evolution of agent-based hypermedia implementations,. Two of them will be discussed in the next sections.

3.3.1 MEMOIR

While the DLS, like other open hypermedia systems, treats hypermedia links as first class entities, MEMOIR (The Memoir project 2001) promotes to first class another kind of object: the *trail.* A user's trail (Bush 1945, Nicol *et al.* 1995) is the set of actions on documents that they have visited (such as opening the document or scrolling through its content) in pursuing a certain task. By storing trails and matching them, we can provide interesting deduction on users' navigation policies. Thus, MEMOIR lets the user ask questions such as "who else has read this document?" and "what else should I read about his subject?".

Software agents are employed to answer these questions for the user: they do so mainly by mining trail information, by doing automatic keywords extraction and by maintaining user profiles. Answers will thus depend on the selected profile, e.g., by verifying similarity scores of documents or the users' personal profile. Currently, simple profiles have been created to capture the nature of the user's work role, e.g. sales manager or researcher.

3.3.2 PAADS (Personal Agent Information Acquisition and Delivery System)

The design of adaptive hypermedia systems (AHS) (Dattolo and Loia 1997) requires the ability to meet users' expectations at runtime; this need is becoming a crucial issue in more recent hypermedia applications, given the increasing availability of all kinds of electronic information and a more intensive integration of different media. Agent technologies can be fruitfully used to create adaptive systems because they can be easily implemented into a fully modular and flexible architecture.

PAADS (Bailey and Hall 2000) is a web-centered architecture that records user profiles, provides adaptive navigation support to users, and presents recorded profiles to others who can query the data; it introduces the concept of linkbases to this agent-based application domain in order to implement adaptive hypermedia navigation; linkbases are used to add links to those web pages that match the corresponding user model. One advantage of this approach is that additional navigation structures can be added to existing web pages without any actual modification to them.

PAADS is organized in three classes of agents:

- *Personal User Agents* follow a single user, accumulating information about his/her interests and expertise, and storing this data in a user model. This model is then sent to the Knowledge Base Agent.
- *Knowledge Base Agents* act as a central repository for data accumulated by personal agents. The knowledge base agent provides an interface to this data allowing other users to query the information via a web browser.
- *Linkbase Agents*, acting as a distributed link service, maintain a local linkbase and provide access to the data for other agents, allowing them to query the linkbase and add or remove links.

Users can enter data directly into their personal agent, or allow their agent to acquire knowledge about them implicitly. This is achieved by recording the user's trails as they browse through the WWW and by extracting keywords from these pages. PAADS represents a first step towards a complete agent-based framework for adaptive systems.

3.4 Agent-Based Link Integrity in the World Wide Web

A hypertext collection is said to have the property of link integrity (Davis 1995) if links always point to existing documents, i.e. the system does not have any dangling links. The hypermedia community has defined other models of hypertexts, such as Hyper-G (Kappe 1995) or Xanadu (Nelson 1987), which guarantee to maintain link integrity. Applying their approaches on the World Wide Web would (and is going to) require time and a fundamental rethinking of the Web architecture.

WWW servers do not maintain any information about the documents that are currently accessed by users and they do not keep track of documents that are bookmarked by users. In such a context, it is impossible to maintain any form of link integrity, or even offer any form of link-related service.

This problem is further amplified by the nature of publishing itself. Every user is allowed to publish documents simply by storing them in a special directory on the Web server. The very presence of a document in this directory makes it accessible through the URL that specifies its access path, and, symmetrically, if the document is removed all documents referring to that URL will contain a dangling link.

The architecture presented by Moreau and Gray (1998) extends the current WWW architecture to the one displayed in Figure 5 and presents an agent-based model to maintain link integrity in the current WWW.

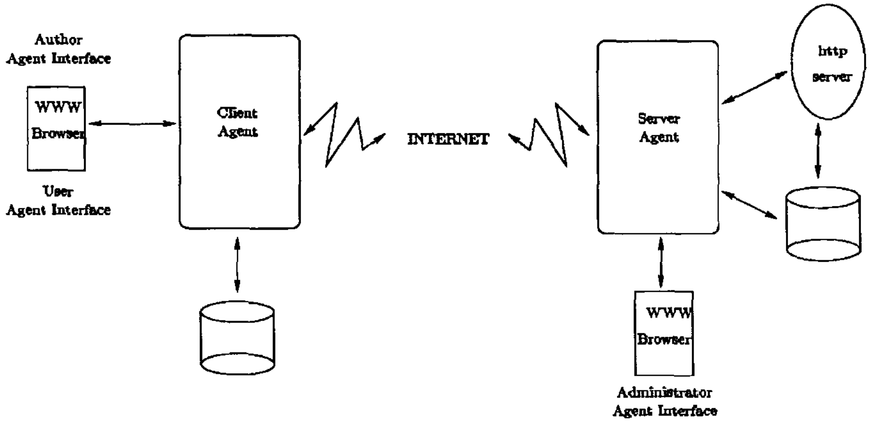


Figure 5. The agent-based architecture.

There are two new essential components, called the *client agent* and the *server agent*, that play an active role in the WWW architecture:

- The client agent is configured as a HTTP proxy on the user's browser so that it can observe, intercept, transform, or redirect all user's requests.
- The server agent acts as a new http server so that it can also observe, intercept, or transform requests, and then can pass them to a regular HTTP server.

Both the server agent and the HTTP server have access to a database of documents. HTTP requests and responses are carried on the Internet as usual; in addition, control messages are exchanged between client agents and server agents. These messages are issued by the client agents whenever the user bookmarks, request the deletion, or forwards via e-mail any document, or by server agents whenever a document is published, updated, or moved. Client agents have to be persistent across sessions, and therefore they also have to have access to a local store.

More precisely, the proposed architecture is distributed and uses the collaboration of three additional agent classes:

- The user agents interacts with the user via the WWW browser. The user agent tries to maintain document accessibility, i.e. link integrity; manages the user's bookmarks, and informs the user of a new version of a document.
- The author agents. The author agents must publish new documents, maintain the existence of previously published documents according to the rules agreed-upon, and inform authors of the current usage of published documents.
- Administrator agents. The administrator agent provides web masters with two new services: they inform administrators about server usage, and facilitate the reorganization of the site while at the same time maintaining link integrity. The administrator agent is a generalization of the log file created by current HTTP servers. Not only does it maintain the access history of the server, but it also provides more dynamic information, such as

users that are currently accessing a document, or users that have bookmarked documents retrieved from this server.

3.5 An Agent-Based Open Hypermedia Model for Digital Libraries

Digital libraries (DL) (Fox *et al.* 1995) are complex information workplaces accessible through a wide-area network that are decentralized, interoperable, heterogeneous in media and tools, scalable. Open Hypermedia Systems (OHS) can play an important role in providing a platform for developing digital libraries (Wiil and Legget 1996, Hall *et al.* 1996).

In fact, OHS have the necessary properties for developing effective digital libraries such as extensibility and scalability, and the dynamic and highly interactive nature of hypermedia which suits better the user-centered information workplaces such as digital libraries (Balasubramanian 1995).

Hypermedia Digital Libraries (HDLs) are digital libraries based on a hypermedia paradigm; they differ from other types of DLs, because they explicitly support intuitive, opportunistic browsing strategies for information seeking. In addition to browsing, HDLs will usually support analytical (i.e. query-based) strategies because these are more efficient in large electronic environments. Users in HDLs can employ different information seeking strategies and engage in rich and complex interactions to achieve their goals.

An agent-based HDL model is presented by Salampasis (1997); the model is grounded on the Dexter hypertext reference model presented in Section 2. Based on the idea of software agents the model introduces the idea of a hypermedia agent: hypermedia agents communicate with their peers using an agent communication language, and thus interoperate in an open and distributed hypermedia environment.

The agents defined in the model are directly derived from the concepts of the Dexter model, even though the storage layer is slightly different. While the Dexter model defines the storage layer as composed of collections of three basic entities (atoms, composites and links), this model adds two more organizational entities: primitives and libraries. Although in the Dexter model the structural semantics of these entities is covered by composites, their existence improve the understanding of how information is organized in a HDL. More specifically, first-level aggregates of atoms are called primitives, while composites are aggregates of higher level objects (i.e. other composites and primitives). Figure 6 depicts the different types of agents and their correspondence to the Dexter model.

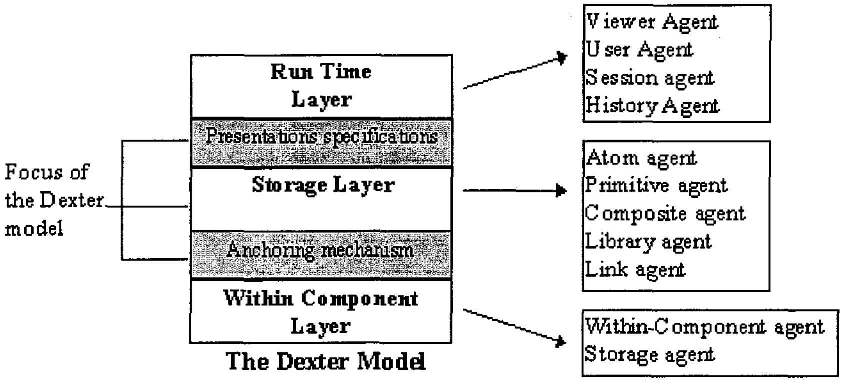


Figure 6. The Dexter model and Corresponding Agents.

Each agent is expected to play a specific role in the digital library:

- Viewer agents provide a viewer to other hypermedia agents;
- Session agent track information about a specific user session and specific instantiations of the hypermedia nodes;
- Atom agents represent raw information (such as text, graphics etc.);
- Primitive agents aggregate related atoms in a single object;
- Composite agents organize hierarchically other components;

- Link agents manage the linking information between components;
- Library agents provide communication with other libraries
- Storage agent provide information storage services (e.g. ftp);
- Within-Component agents handle the internal structure of data (e.g. they may translate from a data format to another one that is recognized by viewer agents).

3.6 The TAO (TeleAction Object) Model

The TAO model (Chang 2000) emphasizes a unified approach to the modeling of multimedia applications, presentation and communication, as a collection of interacting objects. Although the TAO model is not strictly speaking agent-based, we discuss it here because it provides some additional properties to objects that make them more than simple objects, and closer to our view of agents.

The TAO model is composed of a pair: the hypergraph part and the knowledge part. The hypergraph is composed of nodes, composites and links. Bundled nodes can be either basic node or composite, and their sole purpose is presentation. The relations between nodes are defined by using different types of links: attachment link, annotation link, reference link, location link and synchronization link.

The hypergraph determines the priority of the transmission sequence and the presentation order for multimedia objects, according to the currently available communication bandwidth, recipient's environment, link types, nodes types, media types.

The knowledge part allows the objects to become active and using different knowledge levels it enables the user to customize the hypermedia and to specify private information. The user can create and modify the private knowledge of a TAO so that the TAO will react automatically to certain events. The knowledge structure of a TAO is an active index (Chang *et al.* 1995), which consists of a collection of index cells (ICs) with behavior similar to that of agents.

The architecture of the multimedia application development system (Chang and Shih 2001) is shown in Figure 7.

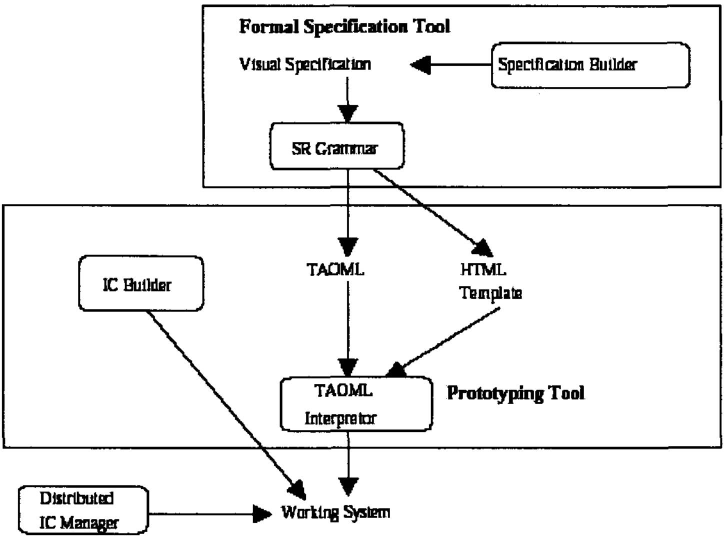


Figure 7. The architecture of the TAO application development.

The development system consists of two tools: the *Formal Specification Tool* allows a specification of the multimedia static specification (MSS) to be created. The specification may be either visual or text-based. The specification is then validated using a Symbol Relation grammar (Arndt *et al.* 1997). If the specification is valid, the tool generates TAOML documents (TAOML is an extension of HTML used to define the TAO nodes), and an HTML template for the specified system. The *Prototyping Tool* includes an Index Cell Builder that creates the knowledge structure of the TAOs. The application generated out of the TAOML documents can then be executed within any web browser communicating with the distributed IC Manager, that controls the active knowledge structure built out of active index cells.

3.7 PageSpace and Coordination Languages

In PageSpace (Ciancarini *et al.* 1998) a different approach to activate the Web is proposed, by redefining the coordination capabilities of the WWW middleware in which the activity takes place.

Client-side technologies such as Javascript, Java or Active-X, and server-side technologies such as CGI, servlets, and server-side include languages such as PHP or ASP provide interesting features for the creation of a more active hypermedia environment in the World Wide Web. These technologies give the ability to "activate" two key components of the WWW architecture, servers and clients. However, we still lack standard and well-known techniques and protocols to allow these components to interoperate. Usually, in fact, projects aiming at the exploitation of the WWW as an active distributed platform locate computing components just on one side (either at the server or at the clients) or, if any more sophisticated need occurs, by inventing an *ad hoc* communication protocol between a specific client and a specific server-side application.

PageSpace is a reference architecture for distributed, coordinable applications working on the World Wide Web. PageSpace defines an *active Web* as a system including some notions of *agent* autonomously performing some activities. These activities can take place at the client, at the server, at the middleware level, at the gateway with another active software system (e.g., an external database, a decision support system or an expert system) or even at the user level. An active Web includes several agents, all with well-defined autonomous behaviors. Each component of an active Web thus is an autonomous agent performing well-defined computations in a shared world providing coordination services: an agent is not just capable of computations but it also should be able to interact (in possibly complex ways) with other agents.

In other architectures the interaction among agents is usually accomplished using client/server architectures (as in any RPC-based system, such as CORBA). However, the client-server framework misses its main goals (for instance, modular design and simple interaction behavior) whenever the interactions among the components is unusually complex, for instance when the components change with time, when the client/server relationship can be reversed, or when the designer needs a wider decoupling among components. A solution to these problems consists in designing the distributed application as a world of agents in which agents are spatially scattered and act autonomously: this schema fits quite well into the distributed objects model.

PageSpace proposes a coordination-based approach where agents perform sequences of actions which are either method invocations or message deliveries. Synchronization actions (e.g., starting, blocking, unblocking, and terminating an activity) are the remaining mechanisms in object invocation. More precisely, there is a clear distinction between *agent computation*, which is what concerns the internal behavior of an agent, and *agent coordination*, which is what concerns the relationship between an agent and its environment, such as synchronization, communication, and service provision and usage.

Coordination models separate coordination from computation, not as independent or dual concepts, but as orthogonal ones: they are two dimensions both necessary to design agent worlds. A coordination language should thus combine two languages: one for coordination (the inter-agent actions) and one for computation (the intra-agent actions). The most famous example of coordination models is the one proposed in Linda [0], which has been implemented on several hardware architectures and combined with different programming languages. Linda can be seen as a sort of assembly coordination language in two ways. First and foremost, it offers very simple *coordinables* (i.e., active and passive tuples, which can be used to respectively represent agents and messages), a unique *coordination medium* (the Tuple Space, in which all tuples reside), and a small number of coordination *primitives*. Second, Linda is a sort of coordination assembly because it can be used to implement higher level coordination languages, such as Jada, a coordination language for Java (Ciancarini and Rossi 1997).

Coordination models can be used to design an architecture that implements a more active view of the Web, where Web activities are not constrained to be neither client-side nor server-side.

The PageSpace architecture (Ciancarini *et al.* 1998) is a proposal for providing a general framework for active agents within a Web environment. In the PageSpace reference architecture, therefore, we distinguish several kinds of *agents*, as illustrated in Figure 8:

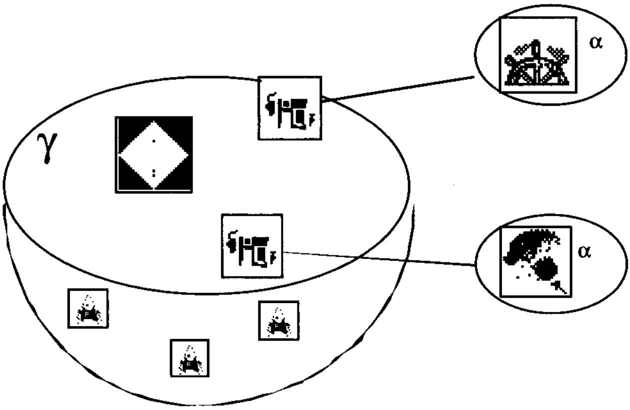


Figure 8. The architecture of PageSpace.

- *User interface agents* (also known as *alpha agents*) are the interfaces of applications. They are manifested as a display in the users browser and are delivered to the client by the other agents of the application according to the requests of the user. Depending on the complexity of the application and the capabilities of the user's browser, there may be different instantiations of user interface agents (in HTML, JavaScript, Java, etc.) that are displayed or executed on the browser.

- *Homeagents* (also known as *beta agents*) are a persistent representation (*avatar*) of users in the PageSpace. Since at any moment users can be either present or absent in the shared workspace, it is necessary to collect, deliver, and possibly act on the messages and requests of the other agents. The homeagent receives all the messages bound to the user, and delivers them orderly to the user on request. Evoluted homeagents can in some circumstances actively perform actions or provide answers on behalf of the user in her absence.
- The *coordination architecture* (also known as *gamma*)is not an agent, but the operating environment, a shared workspace, where the agents live and communicate. Different coordination architectures may provide different capabilities and, ultimately, a different paradigm for creating the agents of the application.
- *Application agents* (also known as *delta agents*)are the agents that actually perform the working of the coordinated application. They are specific of one application, and can be started and interrupted according to the needs of the application. They live and communicate on the coordination architecture, offer and use each other's services, interact with the shared data, and realize useful computations within the PageSpace.
- *Gateway agents* (also known as *epsilon agents*)provide access to the external world for PageSpace applications. Applications needing to access other coordination environments, network services, legacy applications, middleware platforms, etc., may do so by requesting services to the appropriate gateway agent.

Kernel agents (also known as *zeta agents*)provide sensible services to the application agents. They perform management and control task on the agents active within the PageSpace environment. They deal with the activation, interruption and movement of the agents within the physical configuration of connected nodes. Kernels maintain the illusion of a single shared PageSpace when it is actually distributed on several computers, and provide mobility of the agents on the different machines for load balancing and application grouping as needed.

The entities present in the PageSpace architecture can be defined "agents" since they are more than pure objects: the application agents are autonomous and can be active, homeagents work on behalf of the user, etc.

4 Conclusion

The traditional node/link model of hypermedia has shown its limits long ago. Many hypermedia models have been proposed, and several implementations realized, that go beyond this simple schema. The Dexter Reference Model is the most important of these proposals.

The Dexter model, on the other hand, while providing a common vocabulary for the comparison of different implementations, has been surpassed by further advance in the hypermedia field. The World Wide Web, given its availability, and the simplicity of its underlying architecture, has been taken as the most important testbed for advancements in the hypermedia models.

One such advance is agent-based hypermedia, that adds to the basic concepts the concept of agent, an autonomous entity that provides sophisticated behavior to and with the nodes and link.

Whether they are introduced to enable cooperative work, to verify the integrity of links, or to implement active distributed applications, agents share the same common characteristics: they are complex entities with a state, some behaviors, some autonomy, some goals.

Acknowledgments

We wish to thank CNR-IAT and Microsoft Research UK for their partial support to this work.

References

Agha, G. (1986), *Actors: a Model of Concurrent Computation in Distributed Systems*, MIT Press, Cambridge, MA.

Akscyn, R.M., McCracken, D.L., and Yoder, E.A. (1988), "KMS: a distributed hypermedia system for managing knowledge in organizations," *Communications of the ACM*, **31**:7, pp. 820-835, July.

Arndt, T., Cafiero, A., and Guercio, A. (1997), "Multimedia languages for teleaction object,". *Proceedings of 1997 IEEE Symposium on Visual Languages*, pp. 318-327, September.

Bailey, C. and Hall, W. (2000), "An agentbBased approach to adaptive hypermedia using a link service," in Brusilovsky, P., Stock, O., and Strapparava, C. (eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000*, Trento, Italy, pp. 260-263, August, Springer-Verlag.

Balasubramanian, V. (1995), "A hypermedia approach to digital libraries: review of research issues. *SIGLINK newsletter*, **4**:2 pp. 26-28.

Balzano, W., Dattolo, A., and Loia, V. (2000), "An open approach to distribution, awareness and cooperative work," *Proceedings of the Ninth International Conference on Artificial Intelligence: Methodology, Systems, Applications, AIMSA 2000*, September 20-23, Varna, Bulgaria, *Springer-Verlag's Lecture Notes in Artificial Intelligence* – **LNAI**(1904), pp. 122-131.

Berners-Lee, T., Cailiau, R., Luotonen, A., Nielsen, H.F., and Secret, A. (1994), "The World Wide Web," *Communications of the ACM*, **37**, pp. 76-82.

Bieber, M., Vitali, F., Ashman, H., Balasubramanian, V. and Oinas-Kukkonen, H. (1997), "Some hypermedia ideas for the WWW," *Proc. of the 30th Annual Hawaii Int. Conf. on System Sciences*, Wailea, Hawai'i, pp. 309-319, January, IEEE Press.

Brusilovsky, P. (1996), "Methods and techniques of adaptive hypermedia," *From User Modeling and User Adapted Interaction*, **6**:2-3, pp. 87-129.

Bush, V. (1945), "As we may think," *The Atlantic Monthly*, 176:1, pp. 101-108.

Carr, L., De Roure, D., Hall, W., and Hill, G. (1995), "The distributed link service: a tool for publishers, authors and readers," *World Wide Web Journal*, **1**:1, pp. 647-656.

Carriero, N. and Gelernter, D. (1992), "Coordination languages and their significance," *Communcations of the ACM*, **35**:2, pp. 97-107.

Chang, H., Chang, S.K., Hou, T., and Hsu, A. (1995), "Tele-action objects for an active multimedia system," *Proc. of 2nd Inter. IEEE Conf. on Multimedia Computing and Systems*, Washington, D.C., pp. 106-113, May.

Chang, S.K. and Shih, T.K. (2001), *Multimedia Software Engineering*, Handbook on Software Engineering & Knowledge Engineering, World Scientific, vol. 2.

Chang, S.K. (2000), *Multimedia Software Engineering*, Kluwer Academic Publishers.

Ciancarini, P. and Rossi, D. (1997), "Jada: coordination and communication for Java agents," in Vitek, J. and Tschudin, C. (eds.), *Mobile Object Systems: Towards the Programmable Internet*, **LNCS** 1222, pp. 213-228, Springer.

Ciancarini, P., Tolksdorg, R., Vitali, F., Rossi, D., and Knoche A. (1998), "Coordinating multiagent applications on the WWW: a reference architecture," *IEEE Trans. on Software Engineering*, **24**:5, pp. 362-375.

Dattolo, A. and Loia, V. (1997), "Active distributed framework for adaptive hypermedia," *International Journal of Human-Computer Studies*, **26**, pp. 605-626.

Dattolo, A. and Loia, V. (2000), "A concurrent, distributed model for hypermedia-based information systems," *International Journal of Software Engineering and Knowledge Engineering*, **10**:3, pp. 345-349, June.

Davis, H. (1995), *Data Integrity Problems in an Open Hypermedia Link Service*, PhD Thesis, University of Southampton.

Delisle, N. and Schwartz, M. (1986), "Neptune: a hypertext system for CAD applications," *Proceedings of SIGMOD'86*, ACM Press, pp.132-143.

Ferber, J. (1999), *Multi-Agent Systems.* Addison-Wesley: Reading, MA.

Fox, E., Akscyn, R., Furuta, R., and Legget, J.J. (1995), "Digital libraries," *Communications of the ACM*, **38**:4, pp. 23-28.

Garzotto, F., Mainetti, L., and Paolini, P. (1995), "Hypermedia design, analysis, and evaluation issues," *Communications of the ACM*, **38**, pp. 74-87.

Gasser, L. (1991), "Social conceptions of knowledge and action: DAI foundations and open systems semantics," *Artificial Intelligence*, **47**, 107-138.

Gronbæk, K. and Trigg, R.H. (1994), "Design issues for a Dexter-based hypermedia system," *Comm. of the ACM*, **37**, pp. 40-49.

Gronbæk, K. and Trigg, R.H. (1999), *From Web to Workplace: Designing Open Hypermedia Systems*, The MIT Press.

Halasz, F.G. and Schwartz, M. (1994), "The Dexter hypertext reference model," *Communications of the ACM*, **37**, pp. 30-39.

Halasz, F.G. (1988), "Reflections on Notecards: seven issues for the next generation of hypermedia systems," *Communications of the ACM*, **31**:7, pp. 836-852, July.

Hall, W., Davis, H., and Hutchings, G. (1996), *Rethinking Hypermedia: the Microcosm Approach*, Kluwer Academic.

Hardman, L., Bulterman, D.C.A., and Van Rossum, G. (1994), "The Amsterdam hypermedia model: adding time and context to the Dexter model," *Communications of the ACM*, **37**, 50-62.

Hewitt, C. (1991), "Open information systems semantic for distributed artificial intelligence," *Artificial Intelligence* **47**, 79-106.

Kappe, F. (1995), "A scalable architecture for maintaining referential integrity in distributed information systems," *J. of Universal Computer Science*, **1**:2, pp. 84-104.

Lange, D.B. (1990), "A formal model of hypertext," *NIST Hypertext Standardization Workshop*, pp. 145-166, February.

Maes, P. (1994), "Agents that reduce work and information overload," *Communications of the ACM*, 37:7, pp. 31-40, July.

Malone, T.W. and Crowston, K. (1994), "The interdisciplinary study of coordination," *ACM Computing Surveys*, 26:1, pp. 87-119.

Moreau, L. and Gray, N. (1998), "A community of agents maintaining link integrity in the World-Wide Web (preliminary report)," in Nwana, H.S. and Ndumu, D.T. (eds.), *Proceedings of the 3rd International Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-98)*, London, UK.

Nelson, T.H. (1987), *Literary Machines.* Project Xanadu.

Nicol, D., Smeaton, C., and Slater, A.F. (1995), "Footsteps: trail-blazing the Web," *Proc. of the 3rd International World Wide Web Conference*, Darmstadt, Germany, April.

Osterbye, K. and Wiil, U. (1996), "The flag taxonomy of open hypermedia systems," *Proc. of the Seventh ACM Conference on Hypertext*, Washington D.C, USA, pp. 129-139.

Pearl, A. (1989), "Sun's link service: a protocol for open linking," *Proceedings of Hypertext'89*, ACM Press, pp. 137-146.

Rizk, A. and Sauter, L. (1992), "Multicard: an open hypermedia system," *Proceedings of ECHT'92*, ACM Press, pp. 4-10.

Romano Jr., N.C., Nunamaker, J.F., and Briggs, J.R.O. (1997), "User driven design of a Web-based group support system," *Proc. of the 30th Annual Hawaii Intern. Conf. on System Sciences – HICSS30*, Wailea, Hawai'i, Jan. 7-10, vol. II, pp. 366-375.

Salampasis, M. (1997), "An agent-based open hypermedia model for digital libraries," *Proc. of the 3rd Workshop on Open Hypermedia Systems, Hypertext '97*, Southampton, England, April 6-11.

Schlichter, J., Koch, M., and Bürger, M. (1997), "Workspace awareness for distributed teams," in Conen, W. (ed.), *Proc. of the Workshop on Coordination Technology for Collaborative Applications*, Singapore, **LNCS**.

Sohlenkamp, M., Fuchs, L., and Genau, A. (1997), "Awareness and cooperative work: thePOLITeam approach," *Proc. of the 30th Annual Hawaii Intern. Conf. on System Sciences – HICSS30*, Wailea, Hawai'i, Jan. 7-10, vol. II, pp. 549-558.

Stotts, P.D. and Furuta, R. (1989), "Petri-net-based hypertext: document structure with browsing semantics," *ACM Transactions on Information Systems*, **7**:1, pp. 3-29, January.

Streitz, N., Haake, J., Hannemann, J., Lempke, A., Schuler, W., Schütt, H., and Thüring, M. (1992), "SEPIA: a cooperative hypermedia authoring environment," *Proceedings of ECHT'92, ACM Press*, pp. 11-22.

The Memoir project (Managing Enterprise Multimedia Using an Open Framework for Information Reuse) Esprit No. 22153. http://www.mmrg.ecs.soton.ac.uk/publications/Project-MEMOIR.html

Wiil, U.K. (1992), "Issues in the design of EHTS: a multiuser hypertext system for collaboration," *Proceedings of HICSS-25*, IEEE Computer Society Press, pp. 629-639.

Wiil, U.K. and Legget, J.J. (1996), "The HyperDisco approach to open hypermedia systems," *Proc. of Hypertext 96*, ACM press, pp. 140-148.

Wooldridge, M. and Jennings, N.R. (1995), "Intelligent agents: theory and practice," *The Knowledge Engineering Review*, **10**:2, pp. 115-152.

Chapter 5

Self-Organizing Neural Networks Application for Information Organization

R. Rizzo

Document processing is a set of techniques that includes clustering, filtering and retrieval. In the past the interest in these techniques was focused to obtain ranking algorithms for Information Retrieval (IR) systems or effective information filtering algorithms. Today it is growing the need for a new generation of techniques capable to build link relationships between documents or capable to insert the information in the right context. A lot of work has been done in the field of information retrieval but document organization should be more than the ranking methods used by web search engines. So document organization based on semantics is becoming a central issue in information processing in order to help the users to search and browse in large document repositories. Recently some researchers have reported the application of self organizing artificial neural networks in document clustering based on semantics. Using a suitable document representation these techniques can order the document space and generate useful tools to support browsing.

In this chapter after an introduction to document representation the applications of self-organizing networks to document clustering and to information organization are described. Moreover the use of this kind of networks in hypertext development, information filtering and adaptive information systems is also explained.

1 Introduction

The statistics on the growth of the web and on the number of web pages available over the Internet are “always” obsolete. It is difficult, and not necessary, to know exactly the amount of data available over the Internet: the basic concept is that the amount of information in the world is becoming large and there is a need for tools and techniques capable to manage these information. In this scenario information retrieval, organization and filtering are becoming a complex problem, and may be bigger than storage, because computer technologies and hard disk capacity are constantly increasing.

To address these problems it is possible to emulate the associative mechanism used by human mind. In fact when we memorize something we try to link the new pieces of information to something already known. Hypertext and hypermedia technologies are the tool that we used to mimic this process. Moreover using hypertext the author can show more than a set of information: s/he can show the conceptual associations between information atoms, using the link structure to represent semantic relationships between nodes (Allan 1996, Botafogo *et al.* 1992). These associations can also be useful to create a context for the information provided.

Today the large diffusion of the World Wide Web has carried the hypertext in evidence to the audience and nowadays it is impossible to conceive an information repository that is not accessible on the Web. Two of the main consequences of the explosion of the Web are that more and more information are linked with each other and that the information browsing is the most common way to access information. Even the users who use search engines to retrieve information, browse the results and navigate through the web site in order to find the context of the web pages retrieved.

Actually search engines can supply to an user an incredible amount of information, and there is the need for tools that can organize information using document semantics in order to allow the users to effectively browse the search results. Browsing is also fundamental when the desired information is not present in the database so that it needs to be built merging pieces chosen from different documents.

Browsing into a document space of hundreds or thousands of documents is surely an imposing task but it can be made easy if the user is guided within the document set. In order to do that the document space should be ordered in some way; document clustering or taxonomies are the easiest and the most effective way of doing this. Moreover it is useful to remember that an essential part of the process of understanding and learning from the information that come from the outside environment is to build cognitive structures, such as *mental maps* schemes or *networked concepts*.

Artificial intelligence can provide useful and effective algorithms to organize or cluster data in order to help the users building their "mental maps". Recently self-organizing networks have been used to classify information and document in "structures" sometimes two-dimensional graphical representations in which all the documents in a document set are depicted. The documents are grouped in clusters which all concern the same topic, and clusters about similar topics are near each other on the map.

In this chapter the applications to document clustering and to information organization of the self-organizing networks are described. The characteristics of the self-organizing neural networks are briefly introduced and the use of the Self-Organizing Map as a contextual map is reported. After that the vector space document representation and the *document fingerprint* representation are explained and some consideration are reported. The application of self-organizing networks in information organization and visualization are illustrated, and the application to develop an hypertext-like structure are explained. Moreover the use of Self-Organizing Map as information retrieval and filtering tool is reported. The use of SOM map as a component of the user model in an adaptive information system is also explained.

2 Self-Organizing Neural Networks

Our present understanding of biological nervous systems has inspired the artificial neural networks, a dense interconnection of simple non linear computational elements corresponding to the biological neurons. Each connection is characterized by a variable weight that is adjusted, together with other parameters of the net, during the socalled "learning stage". In self-organizing neural networks the elements of the network receive identical input information and compete in their activities. During the *learning stage* each neural unit or group of units is sensitized to a different domain of vectorial input signal values. During the *test stage* this units acts as a *decoder* of that domain.

The SelfOrganizing Feature Map (SOM) is a neural network, proposed by Kohonen (1995), that during the learning stage tries to build a representation of some features of input vectors. This behavior is typical of some areas of the brain where the placement of neurons is sorted and it often reflects some features of sensorial inputs. In the SOM neurons are organized in a lattice, usually one or twodimensional array, that is placed in the input space and is spanned over the input vectors distribution. If the unit array is made by $N_1 \times N_2$ rectangular grid; each unit $h = 1, 2, ...N_1 \times N_2$ has a weight vector $\mathbf{w}_h \in \mathcal{R}^n$ where h define the position of the unit inside the array. Using a two dimensional SOM network it is possible to obtain a map of the input space where closeness between units in the map represents closeness of clusters of input vectors. The main application of SOM is the visualization of complex data in a two dimensional display.

The Growing Neural Gas (GNG) developed by Fritzke (1994) is a self organizing neural network that has no predefined lattice or size

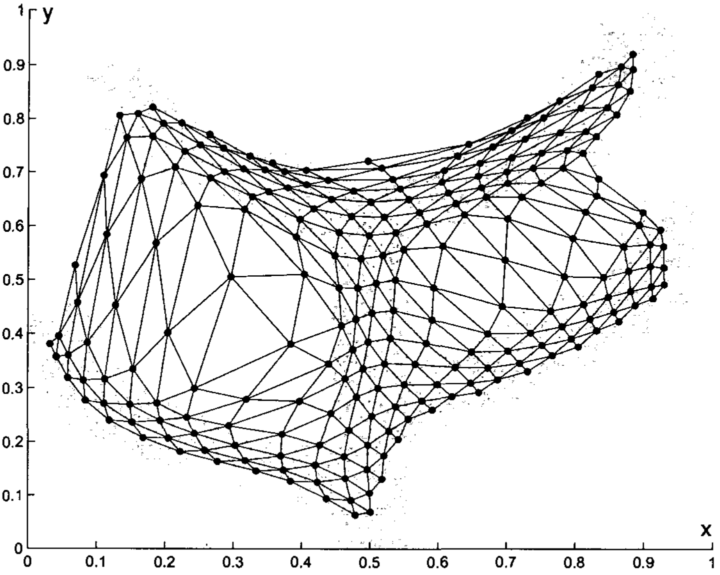


Figure 1. The SOM map approximation of the "cactus" distribution.

and is able to make the topological relations of the input vectors explicit. During the learning stage this network create a graph that can be used to represent the input data distribution. The algorithm can be found in (Fritzke 1994).

The main characteristics that makes GNG network attractive for document organization is that its units are free to represent any input data distribution without a predefined lattice constrain as in the SOM network. This feature can be seen in Figures 1 and 2, where the SOM and GNG networks are used to clustering the same input distribution of (x,y) points. This characteristic is important if the input data has a complex shape (as in Figures 1 and 2) or if it lies in a high-dimensional space.

Another self-organizing network used in documents organization is Adaptive Resonance Theory (ART network). An ART network (Car-

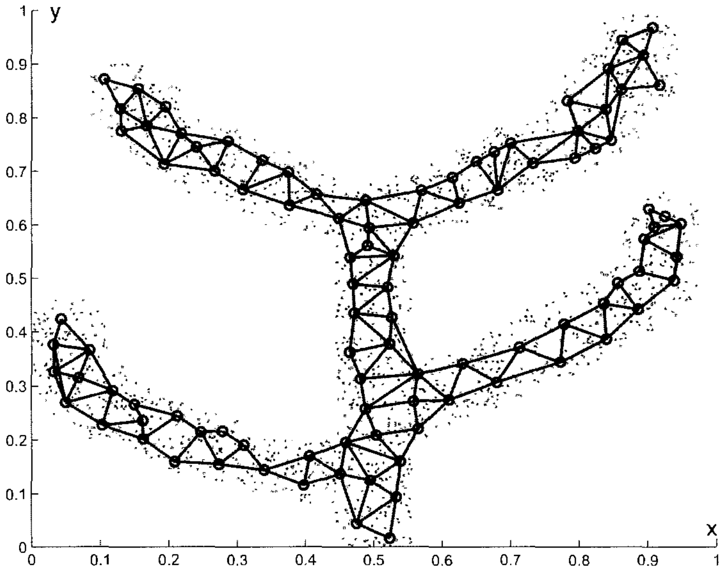


Figure 2. The GNG and the cactus input distribution.

penter and Grossberg 1987, 1998) is capable of clustering arbitrary sequences of input vectors by self-organization. The attempts to categorize a new input by first comparing it with the stored prototypes of existing categories. If no existing matching prototype is found the network considers the input novel and generates a new category. Contrary to the SOM, that has a well defined learning stage, the ART does not have a defined learning stage but continuously attempts to categorize new inputs with the procedure explained above. However the clusters obtained by the ART are not organized in a structure such as a map as they are in the SOM network. In (Merkel 1995b) the result of document clustering using an ART neural network and a SOM map were compared, another application can be found in (Merkel 1995a).

2.1 Contextual Maps

An interesting application of the SOM is related to the organization of *context patterns* where a context pattern is defined as a group of contiguous symbols $S_t = \{s_{i-m}, s_{i-m+1}, s_{i-m+2}, ..., s_{i-1}, s_i, s_{i+1}, s_{i+2}, ..., s_{i+n}\}$ extracted from a symbol string S. A symbol s_i in a symbol string, as a word (or a group of words) in a sentence, has a meaning that depends on its context (the previous symbols and the following ones, in a sentence two or three contiguous words). For the string symbol S_t the context of the symbol s_i is the ordered set $\{s_{i-m}, s_{i-m+1}, s_{i-m+2}, ..., s_{i-1}, s_{i+1}, s_{i+2}, ..., s_{i+n}\}$. Each symbol can be encoded numerically for example by using a random vector, the only requirement is that a metric is definable and that using this metric the distance of a *symbol representation* from itself is zero and the distance between different symbol representations is nonzero and independent from the symbols (Kaski *et al.* 1998). A SOM network can be trained using these representations and it is possible to obtain meaningful "symbols maps" if the context patterns are labeled by their middle symbol and the trained SOM is calibrated using these labels. These maps are used to visualize the relevant relations between symbols according to their roles in their use.

An application to words and sentences is the so-called *Semantic SOM* where a map of the words is created and it is possible to visualize the emergence of words categories from the statistical occurrences of words. These maps are used to obtain a document representation described in the next section.

3 Document Representations

In order to use neural networks to organize document collections, it is necessary to use a vector document representation to obtain the training vectors. In the following paragraphs the so-called *document fingerprint* and the TFIDF representations are described and analyzed in order to highlight their characteristics and their differences.

3.1 The Semantic SOM to Produce the Document Representation

As said before SOM algorithm can be used to obtain contextual maps that visualize a word categorization derived from the statistical occurrences of the words in different contexts. These maps are sometimes called "*word category maps*". In these maps words that appear in the same or similar context are close to each other. In this sense the order of the word on the map seem to follow semantic features and this is the reason why earlier these map were called *Semantic Maps*. These *Semantic Maps* are used to obtain useful information about the word used in a document set and to obtain an effective document representation.

In the seminal paper (Ritter and Kohonen 1989) the SOM was used to sort the words in a collection of texts in order to catch the meaning of the words. An explanation of this method can be found in (Ritter and Kohonen 1989, Kaski 1998).

Each term of a document t_k is coded with a unit-norm vector with random component values $\mathbf{R}(t_k) \in \mathcal{R}^n$ where n can be up to 90. In (Kaski 1998) it was shown that the vectors representing two terms t_i and t_j $\mathbf{R}(t_j)$ and $\mathbf{R}(t_j)$ can be regarded as orthogonal and at least statistically independent.

The context of the term t_k in the document is caught by using the preceding and following word t_i and t_j, and then building a triples of successive words $\{t_i, t_k, t_j\}$, called the "short context" of the t_k word, and represented by:

$$\{\mathbf{R}(t_i), \mathbf{R}(t_k), \mathbf{R}(t_j)\} \tag{1}$$

To speed up the learning stage this "short context" was averaged and the conditional averages

$$E\{\mathbf{R}(t_i)|_{t_k}\} \quad E\{\mathbf{R}(t_j)|_{t_k}\} \tag{2}$$

were used to build the input vector that represent the word t_k and its "average short context" in the text collection.

The vectors $\mathbf{x}_k$ that were used for the learning stage of the semantic SOM are given by:

$$\mathbf{x}_k = \left\{ \begin{array}{c} E\{\mathbf{R}(t_i)|_{t_k}\} \\ \epsilon\mathbf{R}(t_k) \\ E\{\mathbf{R}(t_j)|_{t_k}\} \end{array} \right\}, \mathbf{x}_k \in \mathcal{R}^{3n} \tag{3}$$

where $\epsilon < 1$ (e.g. 0.2) to enhance the influence of the averaged context part over the word part. Due to the average over all of appearances in the texts a word gets the same representation even if it appears in many different contexts or ambiguous situations. After the learning stage the map is labeled by using the following vectors:

$$\mathbf{x}_k = \left\{ \begin{array}{c} 0 \\ \mathbf{R}(t_k) \\ 0 \end{array} \right\} \tag{4}$$

The word category map organizes the words in a two-dimensional array by semantic categories and every unit can be labeled using the words corresponding to the training set vectors. In (Honkela *et al.* 1996, Honkela *et al.* 1995, Kaski *et al.* 1998), many examples of word category map can be found and it is clear that a meaningful order for the word category can be obtained. The similarity between the categories created during self organization is reflected in their distance relationships on the network array, so synonymous terms, or terms which occur in the same context, label the same units or units near each other.

The *word category map* is used to produce a document representation called *document fingerprint* (Honkela *et al.* 1995), a two dimensional histogram obtained by reporting the term frequency value for each word on the word map. A fingerprint for each document in the document set can be obtained. These fingerprints are "blurred" using a Gaussian convolution kernel to reduce its sensitivity to small variations in the documents.

3.2 The TFIDF Document Representation

The Vector Space Representation (VSR) is a common document encoding based on statistical considerations (Salton *et al.* 1994). Using the VSR each document in a document collection is represented by using a vector where each component corresponds to a different word. The component value depends on the frequency of occurrence of the word in the document weighted by the frequency of occurrence in the whole set of documents.

Assuming a set of keyword $\mathcal{V} = \{t_i, i = 1, 2, ..., n\}$ used to represent the documents (i.e. a *dictionary*), each document d_j in a document set $\mathcal{D} = \{d_j, j = 1, 2, ..., m\}$ can be represented as a vector $\mathbf{v}_j$where the element v_{ji} is the weight of the word t_i for that document. This weight can be calculated in many ways: it is possible to simply use the frequency of the keyword t_i in the document d_j or to use more sophisticated mechanisms.

In the so-called TFIDF representation these weights are calculated balancing two different aspects of the keywords frequency:

- the within-document frequency f_{ij} that indicates how many times the term t_i appears in the document d_j;
- the number of documents that contain the keyword t_i;

The aim is to consider "more interesting" a keyword that is repeated in one or few documents and "less interesting" a keyword that is common among the whole set of documents $\mathcal{D}$. According to this analysis the weight v_{ij} can be expressed by the product:

$$v_{ij} = r_{ij} * w_i \tag{5}$$

where the term r_{ij} (relative frequency, TF) takes into account the within-document frequency and can be expressed by:

$$r_{ij} = 1 \tag{6a}$$

$$r_{ij} = f_{ij} \tag{6b}$$

$$r_{ij} = 1 + \log f_{ij} \tag{6c}$$

$$r_{ij} = k + (1 - k) * \frac{f_{ij}}{max_j f_{ij}} \tag{6d}$$

where $max_j f_{ij}$ indicate the maximum frequency of any term in the document d_j. In 6c the first appearance of a term in a document contributes much more than the other occurrences.

The term w_i in eq. 5 is the document-term weight and it reduce the weight v_{ij} if the term appears in many documents of the collection $\mathcal{D}$.

According to the observation of George Zipf (Zipf 1949) the frequency of a word tends to be inversely proportional to its rank. So if the rank is regarded as a measure of the "importance" of the word then the term w_i might be calculated as :

$$w_i = \frac{1}{f_i} \tag{7}$$

where f_i is the number of documents that contain the term t_i. In eq. 7 the term w_i is calculated by using the *inverse document frequency* (IDF). Other, most common, formulation for w_i are:

$$w_i = \log\left(1 + \frac{m}{f_i}\right) \tag{8a}$$

$$w_i = \log\left(1 + \frac{f_{max}}{f_i}\right) \tag{8b}$$

$$w_i = \log\left(\frac{m - f_i}{f_i}\right) \tag{8c}$$

where m is the number of documents in $\mathcal{D}$, and f_{max} is the largest f_{ij} value in the collection.

The eq. 8a is the most common formulation. The logarithm is included to prevent a term for which $f_i = 1$ being considered as twice as important than a term for which $f_i = 2$.

Any of the 6a-d and of the 8a-c, can be used in eq. 5 to calculate the weight v_{ij} and no single combination of them outperforms any other over a range of different queries (Zobel and Moffat 1998).

The calculation of the TFIDF representation often includes a normalization factor that is used to obtain a representation vector that is independent from the text length. The normalized weight can be obtained by using the following formula:

$$w_{ik} = \frac{v_{ik}}{\sqrt{\sum_{i=1}^{n} v_{ij}}} \tag{9}$$

The vector normalization translates the set of points obtained in n-dimensional vectors that are all on the surface of a n-dimensional sphere. This is the reason for using the cosine method to compute the distance between two vectors.

In the following considerations the following representation was used (Balabanovic and Shoham 1995):

$$v_{ij} = \frac{\left(0.5 + 0.5\frac{tf_{ij}}{tf_{max}}\right)\left(\log\frac{m}{df_i}\right)}{\sqrt{\sum_{d_j \in T}\left(0.5 + 0.5\frac{tf_{ij}}{tf_{max}}\right)^2\left(\log\frac{m}{df_i}\right)^2}} \tag{10}$$

where:

tf_{ij} is the number of times the word t_i appears in the document d_j (term frequency),

df_i is the number of documents in the collection which contain the word d_i (document frequency),

m is the number of documents in the document collection,
f_{max} is the maximum term frequency.

3.3 Considerations about the Document Representations

Both the document representations described above share an underlying assumption: they consider a document as a set of words, they do not take into account the relations between words (so that they are often called *bag of word* representations), and do not consider any other attribute of the text. In a document it is not only the sequence of the words which is important: a lot of information in a text is conveyed by the position, size and other attributes like the font of the text (such as bold or italic). An attempt was made to take this information into account in (Molinari and Pasi 1996).

From another point of view the representation of a document can be reconsidered as a "noisy" representation because it is very difficult to completely represent the meaning of a document. This problem is taken into account in the paper (Rauber and Merkel 1998), but there is no effort to "reduce" the noise in the document representation. This "noise" is also due to the fact that a lot of information is contained in a single document and the document is represented in only one vector (the document "fingerprint" or the vector generated by TFIDF), so one item of information becomes "noise" for the others. A simple and intuitive way to reduce this "noise" is to reduce the document length to short pieces of information, ideally to "information atoms," defined in (Ginige *et al.* 1995) as "a piece of information that loses all usefulness if broken down any further." It is difficult to automatically identify concepts within a document but a first approximation could be obtained by separating the document into paragraphs. These are a good starting point because each one has a meaning of its own even if they are sometimes a long way from the definition of an information atom. An attempt in this direction was made by the author in the

Hy.Doc. system (Rizzo *et al.* 1999b).

Moreover it has to be said that the dimensionality of the representation vector in TFIDF representation becomes high if the vocabulary is large, a problem that does not arise if the word category map is used because the *document fingerprint* is always of the same dimension. To overcome this problem, it is also possible to use stem algorithms to reduce the interesting words to their stems, another standard practice in information retrieval (Balabanovic and Shoham 1995).

4 Using a SOM to Organize and Visualize an Information Domain

Today visual interfaces, hypertext technologies and the Internet participate in changing our way to search and to retrieve information and direct browsing on an information set is becoming the most common way to access information. Thus it is necessary to find new methods to organize the information in order to help the user in browsing information. Recently in many papers the applications of self organizing neural networks to document clustering, and in particular of the Self-Organizing Maps, has been emphasized (Chen *et al.* 1996, Orwig *et al.* 1997, Lin *et al.* 1991).

To show this kind of applications the Reuters–22173 document collection can be used. This collection is constituted of 22173 documents of various length, but only 9603 are used in the learning stage in the so called "ModApte split" (Lewis 1991). Many of these documents are manually classified in 10 categories, reported in Table 1. In order to use the SOM network to organize these documents they were represented using the TFIDF with a vocabulary of 400 keywords, the result is the map in Figure 3.

Table 1. The classification of the Reuters collection in the "ModApte split."

Category	Associated number
earn	0
acq	1
money-fx	2
grain	3
crude	4
trade	5
interest	6
ship	7
wheat	8
corn	9

On this map it is possible to distinguish some areas that are populated by documents on a precise topic, so that the map shows clusters of documents and organize them in order to allow the user to browse them. But another important property of the organization induced by the SOM map is that related topic are clustered closely on the map. A user study was conducted by Lin *et al.*(1999) in order to validate the SOM clustering results and in particular this proximity hypothesis. This hypothesis come from the consideration that if two chunks of information are semantically related (for example they are about the same topic) the related representation points in the vector space will be close to each other. To evaluate the associations made by the SOM they were compared to maps generated at random. Concept precision and recall were used as measurement and the following null and alternative hypothesis are taken into account:

- SOM performs no better than a random generated map in terms of precision and recall;
- otherwise (SOM performs better).

Two test collections of documents were used: EBS, a set of brainstorming outputs containing 206 short document, and ITO, a set of

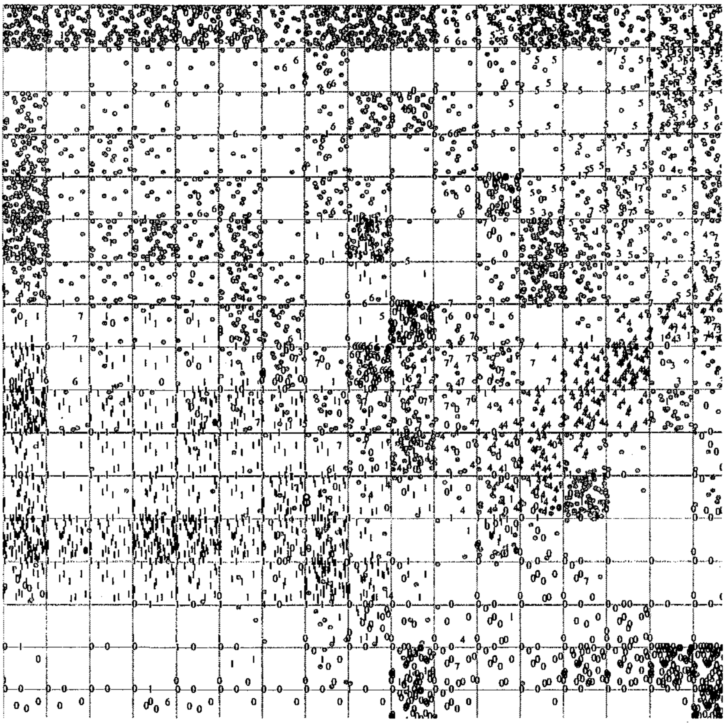


Figure 3. The SOM document map for the Reuters collection. The gray dots indicate documents not classified.

586 project summaries each one is 3-4 pages long. Each document was represented by a vector of 100 keywords for EBS collection and 1000 keywords for the ITO collection) and two different maps were created. 30 human subjects were used for the experimental procedure.

Firstly the regions were sampled from the SOM maps. Secondly for each region were recorded the number of its neighborhood regions and their respective labels. Thirdly the human subjects were asked to select from a list of all region labels the same number of concepts the SOM had found relevant to the sample concept.

The performances of the SOM and the random maps were computed using recall and precision defined as:

$$Precision = \frac{|X \cap Y|}{|X|} \quad Recall = \frac{|X \cap Y|}{|Y|}$$

where:
X represents the terms suggested by the human subject,
Y represents the terms suggested by the SOM or the random map.

From what already said $\mid X \mid = \mid Y \mid$ so$R = P$ and the only R can be used. The comparison of the recall/precision levels of SOM and the random map gives that in EBS collection SOM achieved 32,9 % and the random map the 18,2% in ITO collection SOM had 26.5% versus 6.59% for the random map. So that in (Lin *et al.* 1999) the null hypothesis was rejected.

In (Rizzo *et al.* 1999a) the authors have also compared the organization of the information in semantic clusters obtained by using the SOM to the organization imposed by an hypertext author over the same information set.

In a hypertext system the set of nodes can be considered as a set of points spread over an information space and the links as relationships between them. A common assumption is that two nodes are linked together if they are semantically related in some way. Although this is not always true (e.g. it is not true for hierarchical structures that allows the user to reach more nodes starting from an index node), we focus our attention on the links between semantically related nodes.

If an information map is created by using nodes of an existing hypertext, it is possible to assume that the information map will reflect in some way this semantic order. In other words, it is possible to think that linked information in existing hypertext will be on the same place or near each other in the information map, and so it is possible to compare the link structure built by the developer to the node organization imposed by the neural network. The hypertext was a web course on hypermedia, on Internet at http://wwwis.win.tue.nl/2L670/course.zip. This hypertext is made up of 162 nodes and 357 links. The vocabulary consists of about 6500 words, but it was reduced to n=600 by ignoring stopwords and rare words. The training set is made up of 162 vectors $\mathbf{v}_i \in \mathcal{R}^{600}$ obtained using the TFIDF and a 5×8 SOM lattice was chosen. The simulation has been carried out using the SOM–PAK 3.1 simulator. The result can be expressed in terms of "link precision" as:

$$Precision = \frac{|X \cap Y|}{|X|}$$

where:
X is the set of links imposed by the hypertext author
Y is the set of links imposed by the SOM network

In Figure 4 it is possible to see the link structure over the information map. In this picture the gray boxes represent the neural units and contain the hypertext nodes, as it is possible to see in the magnification of the two nodes.

It has been found that 80 links are between nodes that are on the same place on the information map (the same neural unit), and 151 links are between nodes near each other in the 4 neighborhoods of a rectangular lattice, so 231 links (64.7 % of the total link number) are made between information atoms near each other in the information map. But it has to be said that the information map cannot reproduce a hierarchical structure, which is common in hypertext, so that index node can generate links that cross the whole information map; in the hypertext chosen almost 10 % of the links are due to hierarchical structures.

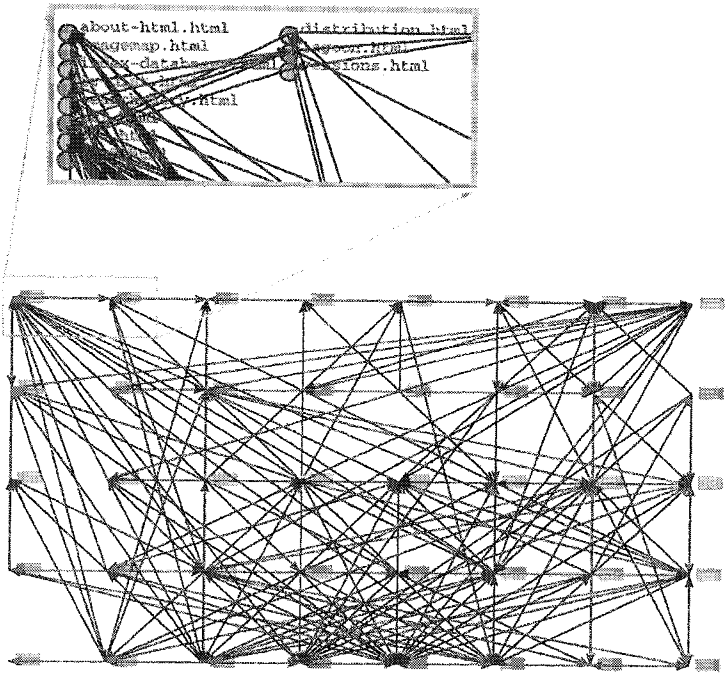
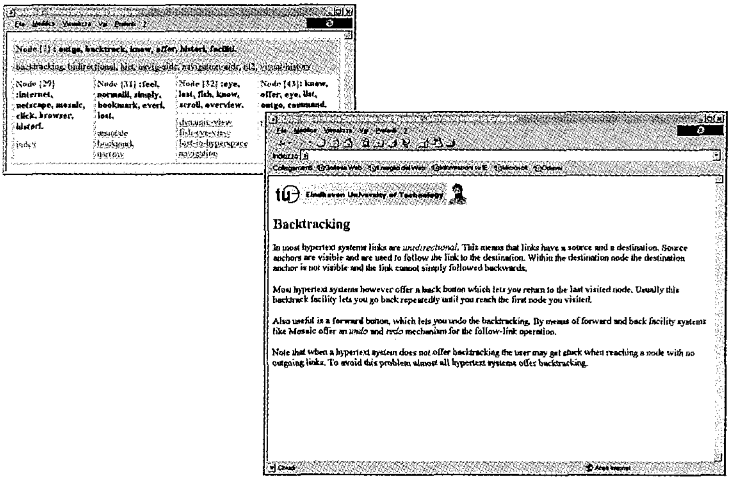


Figure 4. The link structure.

4.1 Other Self-Organizing Networks to Organize the Information Space

The GNG network can be used to group together the documents in clusters and it is possible to see the connections between the neural units as links between document clusters. The major drawback of this approach is that this structure of document clusters and links lays on an n-dimensional space where n is the number of keywords used in the TFIDF representation (600 in the last example). This high-dimensional structure is difficult to visualize. In (Rizzo 1998) the GNG network was trained using the same hypertext system and the generated structure was compared to the original one.

To compare the original hypertext structure to the structure automatically obtained using the GNG network it is necessary to translate the links between document clusters built by the GNG network into links between documents. If a cluster A contains n_A documents and is linked to cluster B that contains n_B documents it is possible to think that all n_A documents in A are linked together and each document in A is linked to each document in B. To test the neural network results the same hypertext was used and the network obtained is composed of 40 linked clusters. Using the above criteria it can be said that the total number of links between documents on the same cluster is 660 of which 78 links are in the original hypertext and the total number of links between documents of linked clusters is 1690 of which 68 are in the original hypertext. The GNG network generates 2356 links between documents of which 146 are in the original hypertext, 40.1 % of the total number of links.


File Modifica Visualizza Vai Preferiti ?
Indirizzo
Collegamenti Galleria Web Il meglio del Web Informazioni su IE Microsoft Outlook
Eindhoven University of Technology

Node [7] : outgo, backtrack, know, offer, histori, faciliti.

backtracking, bidirectional, hist, navig-aids, navigation-aids, ol3, visual-history

Node [29] :internet, netscape, mosaic, click, browser, histori.

index

Node [31] :feel, normalli, simply, bookmark, everi, lost.

annotate
bookmark
narrow

Node [32] :eye, lost, fish, know, scroll, overview.

dynamic-view
fish-eye-view
lost-in-hyperspace
navigation

Node [43]: know, offer, eye, list, outgo, command.

Backtracking

In most hypertext systems links are *unidirectional*. This means that links have a source and a destination. Source anchors are visible and are used to follow the link to the destination. Within the destination node the destination anchor is not visible and the link cannot simply followed backwards.

Most hypertext systems however offer a back button which lets you return to the last visited node. Usually this backtrack facility lets you go back repeatedly until you reach the first node you visited.

Also useful is a forward button, which lets you undo the backtracking. By means of forward and back facility systems like Mosaic offer an *undo* and *redo* mechanism for the follow-link operation.

Note that when a hypertext system does not offer backtracking the user may get stuck when reaching a node with no outgoing links. To avoid this problem almost all hypertext systems offer backtracking.

Figure 5. The user interface to navigate the GNG structure.

Figure 5 shows a simple user interface to navigate the structure created by the GNG network in the information space. The interface is composed of two frames: the little one shows, in the gray area, the content of the cluster (the filenames of the documents that are in the cluster) and below the gray area the content of the clusters linked to it (in this case only 4 clusters are linked to it). The big frame in the foreground shows the document that is obtained by clicking on the file name on the first frame.

The problem of extraction of high-order structure from a GNG network in order to allow visualization is addressed in (Rizzo 2000) where a new neural network structure was developed.

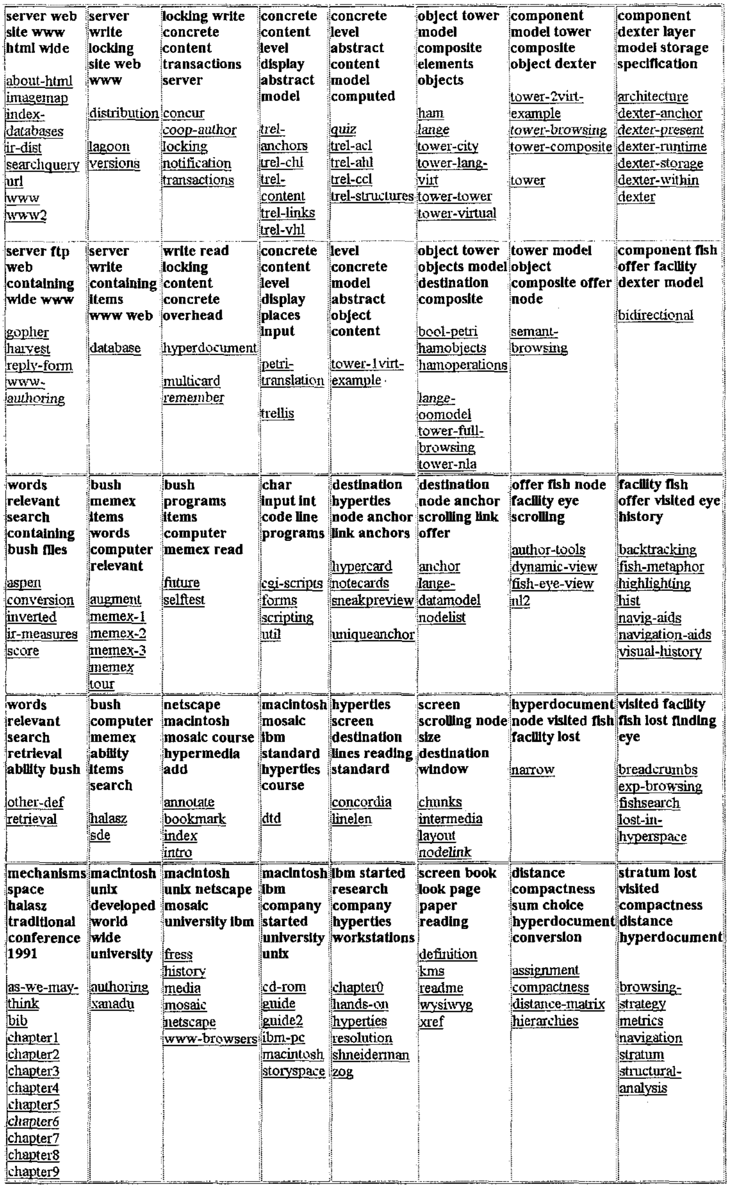
4.2 An Information Map Example

In Figure 6 each neural unit is represented as a box in a 5x8 HTML table that contains one or more hypertext nodes that are semantically related to each other (it really contains a link to the HTML file), and some keywords (in bold) that are generated by taking the six larger vector components of each neural unit. These keywords can give an indication of the kind of documents contained in each box and can help to label some areas in the information map.

These results have suggested us to design and develop a system that adopts the information map to support hypertext-like organization of and access to a set of documents. The system is described in (Rizzo *et al.* 1999a). It should be noted that the adopted HTML-based approach allows the user, by means of a Web-page editor, to move documents from a cell to another one, to change the list of access points, to comment them and so on.

5 Automatic Development of an Hypertext-Like Structure

The same approach used to develop an information map can be generalized to get to a hypertext-like organization of sets of documents. In (Merkel 1998, Roussinov and Ramsey 1998, Kaski *et al.* 1996),



server web site www html wide about-html imagemap index-databases ir-dist searchquery url www www2	**server write locking site web www** distribution lagoon versions	**locking write concrete content transactions server** concur coop-author locking notification transactions	**concrete content level display abstract model** trel-anchors trel-chl trel-content trel-links trel-vhl	**concrete level abstract content model computed** quiz trel-acl trel-ahl trel-ccl trel-structures	**object tower model composite elements objects** ham lange tower-city tower-lang-virt tower-tower tower-virtual	**component model tower composite object dexter** tower-2virt-example tower-browsing tower-composite tower	**component dexter layer model storage specification** architecture dexter-anchor dexter-present dexter-runtime dexter-storage dexter-within dexter
server ftp web containing wide www gopher harvest reply-form www-authoring	**server write containing items www web** database	**write read locking content concrete overhead** hyperdocument multicard remember	**concrete content level display places input** petri-translation trellis	**level concrete model abstract object content** tower-1virt-example	**object tower objects model destination composite** bool-petri hamobjects hamoperations lange-oomodel tower-full-browsing tower-nla	**tower model object composite offer node** semant-browsing	**component fish offer facility dexter model** bidirectional
words relevant search containing bush files aspen conversion inverted ir-measures score	**bush memex items words computer relevant** augment memex-1 memex-2 memex-3 memex tour	**bush programs items computer memex read** future selftest	**char input int code line programs** cgi-scripts forms scripting util	**destination hyperties node anchor link anchors** hypercard notecards sneakpreview uniqueanchor	**destination node anchor scrolling link offer** anchor lange-datamodel nodelist	**offer fish node facility eye scrolling** author-tools dynamic-view fish-eye-view nl2	**facility fish offer visited eye history** backtracking fish-metaphor highlighting hist navig-aids navigation-aids visual-history
words relevant search retrieval ability bush other-def retrieval	**bush computer memex ability items search** halasz sde	**netscape macintosh mosaic course hypermedia add** annotate bookmark index intro	**macintosh mosaic ibm standard hyperties course** dtd	**hyperties screen destination lines reading standard** concordia linelen	**screen scrolling node size destination window** chunks intermedia layout nodelink	**hyperdocument node visited fish facility lost** narrow	**visited facility fish lost finding eye** breadcrumbs exp-browsing fishsearch lost-in-hyperspace
mechanisms space halasz traditional conference 1991 as-we-may-think bib chapter1 chapter2 chapter3 chapter4 chapter5 chapter6 chapter7 chapter8 chapter9	**macintosh unix developed world wide university** authoring xanadu	**macintosh unix netscape mosaic university ibm** fress history media mosaic netscape www-browsers	**macintosh ibm company started university unix** cd-rom guide guide2 ibm-pc macintosh storyspace	**ibm started research company hyperties workstations** chapter0 hands-on hyperties resolution shneiderman zog	**screen book look page paper reading** definition kms readme wysiwyg xref	**distance compactness sum choice hyperdocument conversion** assignment compactness distance-matrix hierarchies	**stratum lost visited compactness distance hyperdocument** browsing-strategy metrics navigation stratum structural-analysis

Figure 6. The HTML table obtained from a SOM document map.

an SOM network has been used to produce, starting from a set of documents, an ordered document map where a user can navigate and find the right document using explorative search. The developed document map has almost the same appearance than the information map shown in Figure 6; but whole documents instead of single information atoms are clustered. However, in order to identify hypertext links starting from a collection of documents (like scientific papers or technical reports), a further step is necessary. In fact, documents are not information atoms, they do not carry a unique idea or concept, they have an introductory part, they have to explain the fundamental ideas and describe the new ones and finally they have a conclusion. In short, they are composed of many information chunks, and it is necessary to break down each document into information atoms.

It is really difficult to separate information atoms starting from a complex document as a scientific paper. For our purposes it has been assumed that paragraphs have been considered as the information atoms. In our system links are generated between the paragraphs of the documents; classification of paragraphs is expected to be much more precise than classification of the whole documents. But it should be noted that, since a map of paragraphs can be misleading and difficult to be read for the end user, a map of documents is also developed and visualized in order to support browsing of the document set.

5.1 The Prototype of the System

The proposed system, called Hy.Doc. creates the link structure by a SOM neural network applied to the paragraphs. For example, as shown in Figure 7, if the user is looking at an interesting idea described in paragraph 2 of document 1, the system will answer by proposing documents 2, 3 and 4 that contain one or more paragraphs related to the one the user is reading (paragraph 3 of document 2, par. 2 of document 3 and par. 2 of document 4). The links generated by the system connect a paragraph to many documents.

However, since browsing between single paragraphs out of their context (the whole document) can be misleading, the links structure between paragraphs is transparent to the user: if the user is reading a specific paragraph, the Hy.Doc. provides him/her the links to all the documents containing paragraphs in the same cluster, rather than to the single paragraphs.

As presented before a SOM map groups document that are semantically related; the classification is more error prone for the reason explained before, but it can give some help. The HTML table has been chosen as a visual representation of a bookshelf, that is an effective metaphor for this kind of organization.

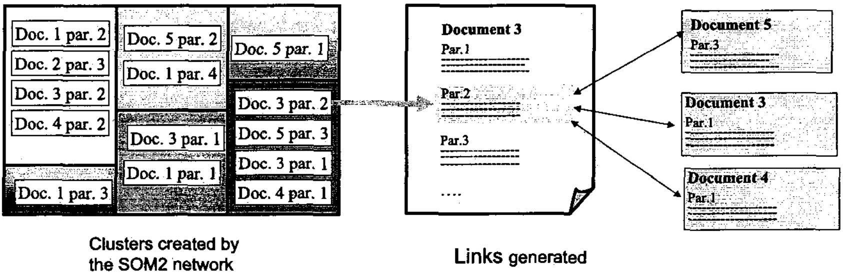


Figure 7. The links between document paragraphs generated by the SOM network.

A user can access the system through an Internet browser. When a user gets access to the system, the document map is sent by the server and visualized in the user browser. S/he can locate the area of interest on the map, choose a document and visualize it by "pointing and clicking" on the map (when a document is visualized, only its location is shown on the map). Afterwards, the user can select a paragraph of interest from the document and ask the Hy.Doc. system for the other documents that contain paragraphs related to it, which are then visualized on the map (Figure 8). The user can look at the topic areas of the returned documents and decide whether they are of interest for him/her or not; the abstract of the documents can be requested in order to support this decision. Implementation details of the system can be found in (Rizzo *et al.* 1999b).

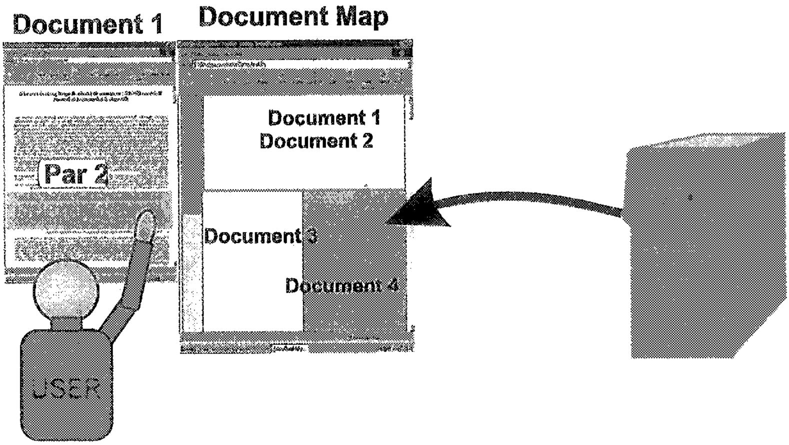


Figure 8. The Hy.Doc. system returns the related documents.

6 Self-Organizing Networks as Information Retrieval and Filtering Tool

When a self-organizing network is used to clustering a set of input vectors $\mathbf{v} \in \mathcal{R}^n$ the input space $\mathcal{R}^n$is divided in Voronoi regions. If set of weight of the SOM network is $W = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{N_1 \times N_2}, \mathbf{w}_i \in \mathcal{R}^n\}$ the input space is partitioned into $N_1 \times N_2$ regions V_h each of them associated with a weight $\mathbf{w}_h$ and defined as:

$$V_h = \{\mathbf{v} \in \mathcal{R}^n \mid ||\mathbf{v} - \mathbf{w}_h|| < ||\mathbf{x} - \mathbf{w}_j||, \forall h \neq j\} \qquad (11)$$

This segmentation of the information domain is useful because a Voronoi region or a set of Voronoi regions, can be considered as an "area" that contains information on the same topic. A consequence of this segmentation is that a document map can be used to answer a user query or to filter documents retrieved over the Internet on the basis of a user query. A query can be considered as a document, so that it is possible to translate it into a representing vector using the TFIDF representation and to classify this query using the SOM in the test stage. If the vector representing the query is given to the SOM during the test stage it will answer with the position of the map in which the network *classifies* the query. This allows the user to receive a list of documents, the documents in that position on the map.

In more formal terms the user query q can be considered as a document of the collection and represented as a vector $\mathbf{v}_q$ using the TFIDF representation. The unit of the SOM network $\mathbf{w}_{h^*}$ for which results:

$$||\mathbf{v}_q - \mathbf{w}_{h^*}|| < ||\mathbf{v}_q - \mathbf{w}_j||,\ h^* \neq j,\ j = 1, 2, ..., N_1 \times N_2 \qquad (12)$$

is called the *best matching unit* (bmu) and the documents in the Voronoi region V_{i^*} are the retrieved documents.

Moreover if a user gets a set of documents by a search engine, it is possible to use the SOM network to find their right places in the bookshelf and rank the retrieved documents according to their distance from the query. As an example, in Figure 9 the document number 1 will be ranked as relevant because is near the position of the query "position" on the map, on the converse the document number 2 will be considered not relevant.

6.1 The EDGES System

The organization of a huge set of documents, in order to serve a community of people with different interest and information needs cannot be afforded by using a single neural network. An effective approach is to used many neural networks that organize a subset of the information domain. In (Merkel 1998, Merkel and Rauber 1999) a hierarchical neural network structure was proposed. A drawback of this approach is that all the documents, and the hierarchical SOM structure, are in the same high-dimensional vector space. The approach

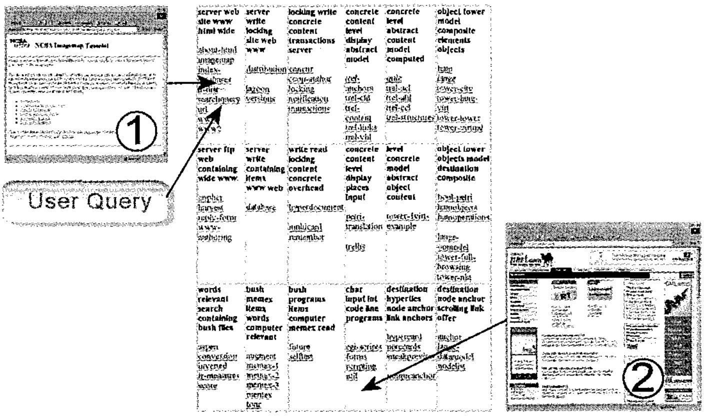


Figure 9. The SOM map used as information filtering tool.

used in EDGES (EDucational aGEnt Server) (Rizzo *et al.* 1998) exploits the characteristics of the intelligent agent technology in order to build an intelligent interface for the SOM map and to obtain a system scalable and capable to serve users with different interests. In this application the different document sets are organized by many SOM map using the fittest vocabulary and the best TFIDF representation.

The aim of the EDGES (EDucational aGEnt Server) project is to build a set of servers that allow the user to access information on a specific topic, to filter the documents obtained using a search engine and to build a document database, using an agent based computer interface. A prototype of the system was developed using $Java^{TM}$ and aglets technology (Lange 1997).

The EDGES system is composed of servers that contain the following components:

- A set of *master agents*, one for each user. A master agent helps its user to obtain the required information: it gets the query from the user and create the search agent to look for the desired infor-

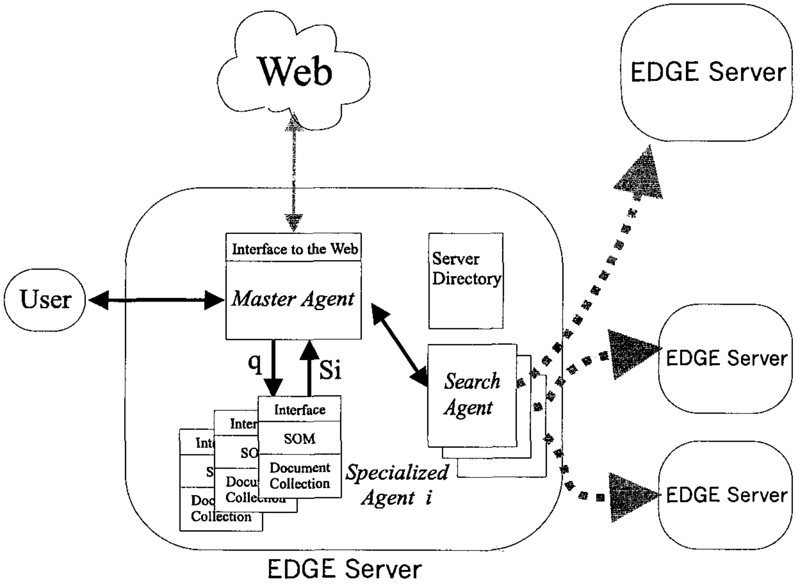


Figure 10. The structure of an EDGE server.

mation in other EDGE servers; moreover, the master agent can create the user interface and present the search results to the user. On user demand, it can query the Web using a search engine and distribute the search results to the specialized agents for filtering.

- *Search agents*, dynamically created by the master agents; they deliver the user query to the EDGES servers. They also deliver the URL of documents to be filtered.
- A set of *specialized agents*, one for each topic. A specialized agent manages a document database through a SOM network. It answers to the query carried by a search agent finding the best matches to the query and proposing them to the user, together with a set of keywords and some annotations if available. Periodically, the specialized agent reorganizes the database and repeats the learning stage of the neural network. Finally, by using the neural network, it can filter the documents retrieved by a search engine and propose only the best matches to the query.

- A directory of other available EDGES servers, used by the master agents and the search agents to know the addresses of the other EDGES servers.

If a user wishes to retrieve information on a particular topic, s/he can ask to her/his master agent to search this information. The master agent does not have to know where the fittest specialized agent is, since it passes the query q to the local specialized agents and wait for an answer.

The query q constitutes the most significant part of a message between the master agent and the specialized agent. Suppose that in a EDGE server are k specialized agents, they receive the query q and build its representations $\mathbf{v}_q^1, \mathbf{v}_q^2, ...\mathbf{v}_q^k$ in different spaces with different dimensions using the eq. 10.

The SOM networks managed by each specialized agent will respond with the activation of the *bmu* units: $\mathbf{w}_{bmu}^1, \mathbf{w}_{bmu}^2, ...\mathbf{w}_{bmu}^k$. and each agent i will respond with the set $S^i = \{r_d^i | \mathbf{v}_d^i \in V_{bmu}^i\}$ of the references of documents r_d^i which representing vectors $\mathbf{v}_d^i$ are inside the Voronoi regions of the *bmu* unit $V_{bmu}^i, i = 1, 2, ..., k$.

If no one of the local agents can satisfy the query (i.e. no one of the local specialized agents knows the query topic), the master agent forwards the query to the other EDGES servers on the directory using search agents. When each search agent reaches a new EDGES server, it sends the user request to the local specialized agents and wait for the answer. If a specialized agent can satisfy the request, it sends the URL of the best-matched documents back to the search agent that, in turn will return them to the master agent. If the specialized agent cannot satisfy the request, the search agent will read the local directory of other EDGES servers, will clone itself to reach the servers not yet visited and will die. The master agent will show the answers of the specialized agents to the user and can query a search engine if the user wants it. The URLs returned by the search engine can be filtered using the user query and the knowledge of the right specialized agent.

7 Overlay User Model by Using a SOM

Adaptive systems are those systems that are capable to change their behavior and its functionality in order to meet the user requirements (Brusilovsky and Eklund 1998). These systems need a model of the goals, preferences and knowledge of each user. A SOM network can be used to build a map of the information domain in an hypermedia adaptive learning system. Using this map it is possible to build an overlay model of the knowledge of the user; in another layer it is possible to build the map of the learning goals. This structure was proposed in the system DAWN(Designing in Architecture Working on the Net), a Webbased instruction system aimed at supporting new didactic approaches to the subjects of Urban Planning and Architecture (Fulantelli *et al.* 2000).

In the DAWN system the information space is sorted in a SOM document map according to their semantic content; this map, which is part of the Information Domain Model, is shown in the upper left side of Figure 11. The new documents added to the system are automatically sorted by the SOM network, so that the expert is not forced to check all the documents at the time when they are introduced into the system, rather they can be checked periodically in order to correct the misclassifications of the SOM algorithm.

By classifying the user's answers to an entry test, the system builds up an *overlay model*of the user's knowledge, with reference to the clusters identified by the SOM map and not to single information in the information space (in the middle of Figure 11). An important consequence of this approach is that the model will remain consistent even if new documents are added to the system. The knowledge needed to accomplish the assignments of the course is mapped to the Information Domain map thus producing the Assignment Matrix represented in the lower part of Figure 11. The system allows the expert to map the assignment by directly inspecting the Information Domain Map (represented as an HTML table) and deciding what knowledge level is required to reach the learning goal (knowledge levels are represented as levels of gray in Figure 11). However, this solution has a drawback: if the mapping of the assignment is left entirely to the system then the misclassification errors should be the same in the Information Domain mapping, and in the User Knowledge mapping. So that the three models will remain consistent.

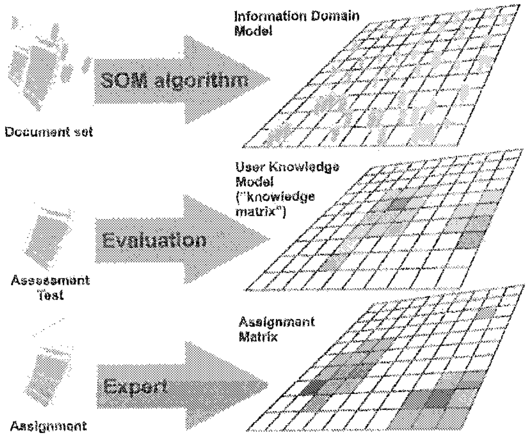


Figure 11. The three models based on SOM maps.

7.1 Adaptive Navigation Support

The solution we propose is based on an adaptive link annotation technique (Brusilovsky *et al.* 1998)that marks all the links on the page and suggests which page an user should visit next according to the learning goal, the semantic value of each node that can be reached and the student's knowledge. The user preferences for approaching the information were also taken into account.

The user knowledge model is based on the student's background knowledge and the knowledge s/he acquires during the navigation towards the learning goal. As stated above the first time an user enters DAWN, s/he is asked to complete an entry test to assess his/her knowledge related to the course subjects. To be more precise the entry test introduces a certain number of questions for every document cluster as identified in the Information Domain Model by the SOM. According to the answers, an automatic procedure produces a representative matrix of the student's background knowledge on the course subjects ("knowledge matrix" in Figure 11). The value in each cell (which represents a cluster in the information domain model) is a number corresponding to the level of knowledge for that subject.

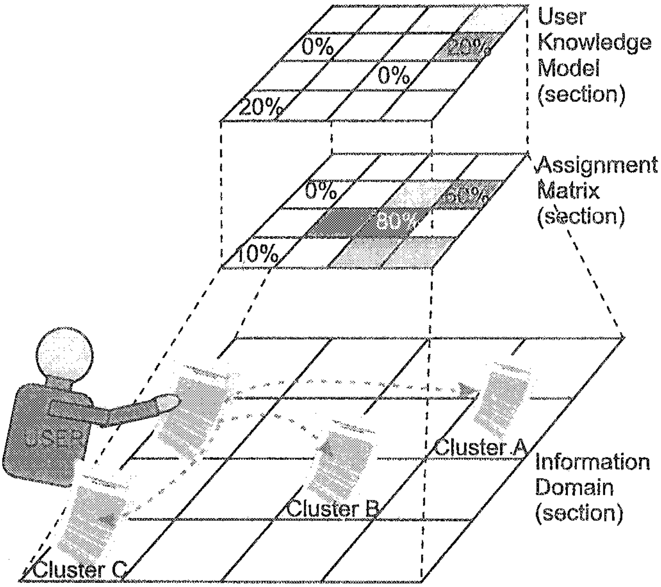

0%
20%
0%
20%
User Knowledge Model (section)
0%
50%
80%
10%
Assignment Matrix (section)
USER
Cluster A
Cluster B
Cluster C
Information Domain (section)

Figure 12. The combination of the Knowledge Matrix and of the Assignment Matrix to annotate links in the DAWN system.

7.2 Link Annotation

In order to annotate the links on each page visited by the user. The system distinguishes between the links that point to documents which are related to the assignment and the links pointing to documents which are not related to. This first classification of the links is based on the difference between the elements of the "Knowledge Matrix" and the "Assignment Matrix": the system evaluates the level of student knowledge for each cluster in the "Knowledge Matrix" (for example expressed in percentages as shown in the upper section of Figure 12) and the level of knowledge for each cluster that the student has to acquire to achieve the learning goal, as reported in the "Assignment Matrix" (as shown in Figure 12). Therefore the system evaluates the links pointing to the pages in the clusters A and B as recommended (because they increase the knowledge of the concepts required by the learning goal); then it marks the links to the pages on the cluster C as not recommended.

8 Conclusions and Future Works

Self-organizing networks, such as SOM and GNG, sort documents that are semantically related to each other into clusters and they organize these clusters on a map or they connect these clusters creating a lattice structure on the information space. This organization translates a semantic relationship in a neighboring relationship that can easily be visualized on a map if the SOM network is used.

The lattice structure created between the information clusters connects information semantically related and a straightforward application is the development of a system that builds an hypertext-like structure over a set of documents. This structure connects documents to other documents (or documents fragments to other documents fragments) but more investigations are needed in this direction to obtain a true hypertext structure. Moreover it is a necessary tool to help the user navigation, due to the high number of links generated, especially when a GNG network is used.

The neighborhood relationship between information was used to build an information filtering and retrieval tool. The behavior of the information filter can easily be defined by the user inspecting the information map and deciding which is the interesting topic (i.e. the suitable information cluster). This propriety can be exploited by developing a new kind of visual interfaces that can enhance the user control on the behavior of an intelligent agent.

Another application of the self-organizing networks is the representation of a knowledge domain in a way that make it easy to inspect, to manipulate and to manage such domain. This representation can also be used to build open adaptive learning systems and a prototype of that kind of system was described.

Self Organizing Maps can organize and visualize information in a way that help an user to build a cognitive structure that supports browsing activity and management of a large information space specially when the user is not aware of the information in the space. These cognitive structures are needed in order to avoid the effects of an "information overload."

Acknowledgment

The author would like to thank the AACE for permission to use the document used for the Hy.Doc. system, and his colleagues Allegra M., Fulantelli G., Arrigo M., for their research work, and Munna E. for the programming work.

References

Allan, J. (1996), "Automatic hypertext link typing," *ACM Hypertext 96*, Washington DC, March 1620, pp. 42-52.

Balabanovic, M. and Shoham, Y. (1995), "Learning information retrieval agents: experiments with automated web browsing," *AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Resources*, Stanford, CA, March.

Brusilovsky, P. *et al.* (eds.) (1998), *Adaptive Hypertext and Hypermedia*, Kluwer Academic Publisher.

Brusilovsky, P. and Eklund, J. (1998), "A study of user model based link annotation in educational hypermedia," *Journal of Universal Computing Science*, vol. 4, no. 4, pp. 429-448, Springer Pub. Co.

Botafogo, R., Rivlin, E., and Shneiderman, B. (1992), "Structural analysis of hypertext: identifying hierarchies and useful metrics," *ACM Transactions on Information Systems*, vol. 10, no. 2, April, pp. 142180.

Carpenter, G. and Grossberg, S. (1987), "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics and Image Processing*, vol. 37, pp. 54-115.

Carpenter, G.A. and Grossberg, S. (1998), "The ART of adaptive pattern recognition by a self-organizing neural network," *IEEE Computer*, vol. 21, no. 3, pp. 77-88.

Chen, H., Schuffels, C., and Orwig, R. (1996), "Internet categorization and search: a self-organizing approach," *Journal of Visual Communication and Image Representation Special Issue on Digital Libraries*, vol. 7, no. 1, pp. 88-102.

Fritzke, B. (1994), "A growing neural gas network learns topologies," *NIPS 1994*, pp. 625-632, Denver.

Fulantelli, G., Rizzo, R., Arrigo, M., and Corrao, R. (2000), "An adaptive open hypermedia system on the Web," *International Conference on Adaptive Hypermedia and Adaptive Hypermedia Web-Based Systems, AH 2000*, Trento, Italy, Aug. 28-30.

Ginige, A., Lowe, D.B., and Robertson, J. (1995), "Hypermedia authoring," *IEEE Multimedia*, pp. 24-34, Winter.

Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1996), "Newsgroup exploration with WEBSOM method and browsing interface," Technical Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo. WEBSOM home page (1996) available online at http://websom .hut.fi/websom/.

Honkela, T., Pukki, V., and Kohonen, T. (1995), "Context relations of words in grimm tales analyzed by self-organizing map," *Proc. 5th Int'l Conference on Artificial Neural Networks, ICANN'95*, Paris, Oct. 9-13, vol. 2, pp. 239-244.

Kaski, S., Honkela, T., Lagus, K., and Kohonen, T. (1996), "Creating an order in digital libraries with selforganizing maps," *Proceedings of WCNN'96, World Congress on Neural Networks*, September 1518, San Diego, pp. 814817.

Kaski, S., Lagus, K., Honkela, T., and Kohonen, T. (1998), "Statistical aspect of the WEBSOM system in organizing document collections," in Scott, D.W. (ed.), *Computing Science and Statistics*, vol. 29, pp. 281-290, Interface Foundation of North America Inc.: Fairfax Station, VA

Kaski, S. (1998), "Dimensionality reduction by random mapping: fast similarity computation for clustering," *Proc. of IJCNN'98*, Anchorage, Alaska, May, 4-9, pp. 413-418.

Kohonen, T. (1995), *SelfOrganizing Maps*, SpringerVerlag, Berlin.

Lange, D.B. and Oshima, M. (1997), "IBM Research: Programming Mobile Agents in Java." Available online at http://www.trl.ibm .jp/aglets/agletbook/index.html.

Lewis, D.D. (1991), "Evaluating text categorization," *Proceedings of Speech and Natural Language Workshop*, Defense Advanced Research Projects Agency, Morgan Kaufmann, pp. 312-318.

Lin, X., Soergel, D., and Marchionini, G. (1991), "A self-organizing semantic map for information retrieval," *Proc. of the* 14th *Annual International ACM/SIGIR Conf. on Research and Development in Information Retrieval*, pp. 262-269, Chicago IL, Oct. 13-16.

Lin, C., Chen, H., and Nunamaker, J.F. (1999), "Verifying the proximity hypothesis for self-organizing maps," *Proc. of the* 32nd *Hawaii International Conference on System Sciences*.

Merkel, D. (1995a), "Content-based software classification by self-organization," *Proc. of the IEEE Int'l Conference on Neural Networks, ICNN 95*, Perth, Australia, Nov. 27-Dec. 1, pp. 1086-1091.

Merkel, D. (1995b), "Content-based document classification with highly compressed input data," *Proc. 5th Int'l Conference on Artificial Neural Networks, ICANN'95*, Paris, Oct. 9-13, vol. 2, pp. 239-244.

Merkel, D. (1998), "Text data mining," in Dale, R., Moisl, H., and Somers, H. (eds.), *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, New York: Marcel Dekker.

Merkel, D. and Rauber, A. (1999), "Self-organization of distributed document archives," *Proc. of the* 3nd *Int'l Database Engi-*

neering and Applications Symposium, IDEAS'99, Montreal, Canada, Aug. 2-4.

Merkel, D. and Rauber, A. (2000), "Document classification with unsupervised neural networks," in Crestani, F. and Pasi, G. (eds.), *Soft Computing in Information Retrieval*, Physica Verlag & Co., Germany, pp. 102-121.

Molinari, A. and Pasi, G. (1996), "A fuzzy representation of HTML documents for information retrieval systems," *Proc. of Fifth IEEE International Conference of Fuzzy Systems*, Sept. 8-11.

Orwig, R.E., Chen, H., and Nunamaker, J.F. (1997), "A graphical, self-organizing approach to classifying electronic meeting output," *Journal of the American Society for Information Science*, vol. 2, no. 2, pp. 157-170.

Rauber, A. and Merkel, D. (1998), "Creating an order in distributed digital libraries by integrating independent self-organizing maps," *Proc. of ICANN'98*, Skovde, Sweden, Sept. 2-4.

Ritter, H. and Kohonen, T. (1989), "Self organizing semantic maps," *Biological Cybernetics*, vol. 61, pp. 241-254.

Rizzo, R. (1998), "Self organizing networks to map information space in hypertext development," *Proc. of NC 98*, Vienna, Sept. 23-25, ISBN 3-906454-15-0.

Rizzo, R. (2000), "A neural network tool to organize large document sets," *Proc of AIMSA 2000*, Varna, Bulgaria, Sept. 20-23, pp. 301-309, Lecture Notes in Artificial Intelligence, Springer.

Rizzo, R., Munna, E. and Arrigo, M. (1998), "EDGES server: developing an educational distributed agent system," *Proc. of AACE WebNet 98*, Orlando, Nov. 6-12.

Rizzo, R., Allegra, M., and Fulantelli, G. (1999a), "Hypertext-like structures through a SOM network," *Proc. of ACM Hypertext '99*, pp. 71-72, Darmstadt, Germany, Feb. 21-25.

Rizzo, R., Allegra, M., and Fulantelli, G. (1999b), "Hy.Doc: a system to support the study of large document collections," *Proc. of ICL99 Workshop*, Villach, Austria, Oct. 7-8, ISBN 3-7068-0755-6.

Roussinov, D. and Chen, H. (1998), "A scalable self-organizing map algorithm for textual classification: a neural network approach to thesaurus generation," *Communication and Cognition – Artificial Intelligence*, 15(1-2), pp. 81-112.

Roussinov, D. and Ramsey, M. (1998), "Information forage through adaptive visualization," *Proc. of the 3rd Conference on Digital Libraries*, Pittsburgh, June 23-26, pp. 303-304.

Salton, G., Allan, J., and Buckel, C. (1994), "Automatic structuring and retrieval of large text files," *Communications of ACM*, vol. 37, no. 2, pp. 97-108.

Zipf, G.K. (1949), *Human Behaviour and the Principle of the Least Effort*, Reading, MA: Addison-Wesley.

Zobel, J. and Moffat, A. (1998), "Exploring the similarity space," *ACM SIGIR Forum*, vol. 32, no. 1, pp. 18-34, Spring.

This page is intentionally left blank

Chapter 6

Emotion-Orientated Intelligent Systems

T. Ichimura, T. Yamashita, K. Mera,
A. Sato, and N. Shirahama

"Kansei" engineering is now widely known as a research field of human feeling or sensory systems. An emotion oriented intelligent system includes deeper consideration for human emotions. We feel that an emotion-oriented intelligent system equipped with a human-like interface enables smoother human communications. Aside from verbal messages, human face-to-face communication usually includes nonverbal messages such as facial expressions, tone, speaking rate, pauses, hand gestures, body movements, posture, and so on. Also, we can be deeply impressed by listening to music and watching pictures and movies. We are able to feel similar nonverbal messages from these entities. In this chapter we explain some techniques and systems for understanding human emotions through several test cases, including an e-mail application, an automatic scenario analysis system for Japanese traditional opera, a retrieval system for music databases based on user's impressions, and an artificial emotion processing system based on the relationship between color expressions and human emotions.

1 Overview of Emotion Orientated Intelligent Systems

Science has made great advances in the past century. The scientific or engineering technologies have not always dealt with human emotion explicitly. At the same time, humans who use those technologies do not necessarily have a reasonable behavior, because every human has emotions, is carried away by the emotion and lives everyday life with some feelings.

When we consider the interaction between emotion and cognition from the psychological point of view, the idea, which treats the emotion as an obstructive thing, might be old-fashioned. Since the research field of engineering aims to achieve a smooth communication between humans and machines, it is natural that we wish to develop the technique for understanding human's emotions or to embed the emotional functions in the machines. Therefore, we consider emotionally orientated intelligent systems.

Recently, "Kansei" engineering (Tsuji 1997) has been widely known as a research field of human feeling or sensory. An emotion orientated intelligent system can deeply consider the emotions of humans. We hope that the emotion orientated intelligent system is equipped with human-like interface, which enables human to like communications. Human face-to-face communication is usually giving and taking nonverbal messages such as facial expressions, vocal inflection, speaking rate, pauses, hand gestures, body movements, posture, and so on. Furthermore, we are deeply impressed by listening to music and watching pictures and movies. We shall be able to feel such nonverbal messages.

In this chapter, we introduce some techniques and systems of understanding human's emotions. Sections 2 and 3 describe the e-mail application software Facemail and the automatic scenario analysis system for the Noh play, Japanese traditional opera, as facial expression systems. Section 4 explains the retrieval system from music database based on user's impression. Successively, Section 5 proposes an artificial emotion processing system based on the relation between color expressions and human emotions. Section 6 will shed a new light on the future of the research of intelligent systems based on human emotion.

2 Facemail

First, we developed application software with emotional facial expressions as emotion orientated intelligent systems. It can analyze emotions of the user and represent his/her emotions as facial expressions. The application software has two outstanding functions. One function is displaying emotional faces and the other one is analyzing emotion from sentences. The face displaying part is based on a sand glass type neural network trained by real face images (Ueki *et al.* 1993). The emotion analyzing part is done with Emotion Generating Calculations (EGC) method based on the Elliot's Emotion Eliciting Condition Theory (Elliott 1992). The proposed method can judge whether an event is pleasant or not and can calculate each emotion value for 20 kinds of emotions. These attributes are successfully set to the trained neural network as input signals.

2.1 Parallel Sand Glass Type Neural Networks

We give a brief explanation of a sand glass type neural network in Figure 1, which can learn different teaching signals simultaneously (Irie and Kawato 1990, Fukumura *et al.* 1998). This type of network consists of several neural networks and each network has 5 layers. The third layer in each network is connected with each other. The neurons in the other layers activate independently in each network. The same teaching signals are set into input neurons and output neurons in each network to perform an identity mapping. Figure 1 shows *N* combined networks and it can learn *N* kinds of different training data. The information related to input signals are condensed into the neurons in the third layer.

Back Propagation (BP) learning algorithm is employed to train the network. However, in the third and in the fifth layer, we adopt a linear function as a bias function instead of a sigmoid function, and do not use threshold values to represent prominent weights of its incoming links.

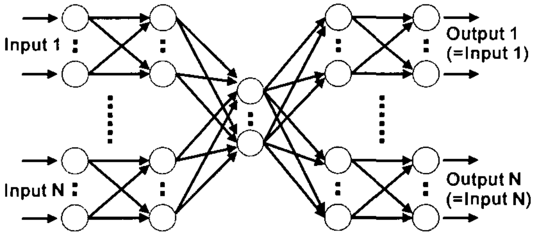


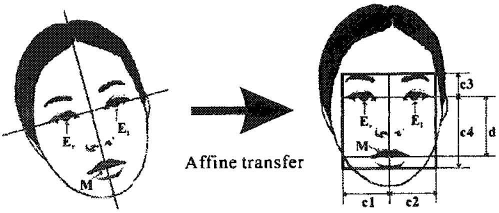
Figure 1. An overview of a parallel sand glass type neural network.

2.2 Facial Training Data

In this chapter we use emotional facial expression of some persons as teaching signals. For each person, there are 6 facial expressions for basic emotions, "happiness," "sadness," "disgust," "anger," "fear," "surprise," and a neutral one. Each emotional face has two face images. Therefore, we have 13 pieces of pictures for each person. In order to construct an emotion space in the third layer in the neural network, we tried to construct a facial expression model by the neural networks.

In order to normalize the face images in position and size by using internal facial features as reference points, we use an affined transformation (Akamatsu *et al.* 1993) to extract the normalized target images. First, we determine three reference points, $\mathbf{E}_r$, $\mathbf{E}_l$, and $\mathbf{M}$ as the center points of the regions that correspond to the both eyes and mouth in Figure 2. In this chapter, the parameters in Figure 2 are $c_1 = c_2 = 0.8d$, $c_3 = 0.4d$, $c_4 = 1.2d$. Then, we obtained a standard window of 128×128 pixels to form the target images. Figure 3 shows the 6 facial expressions for basic emotions and a neutral face of a subject. These pictures are transformed into 8-bit gray-scale format.

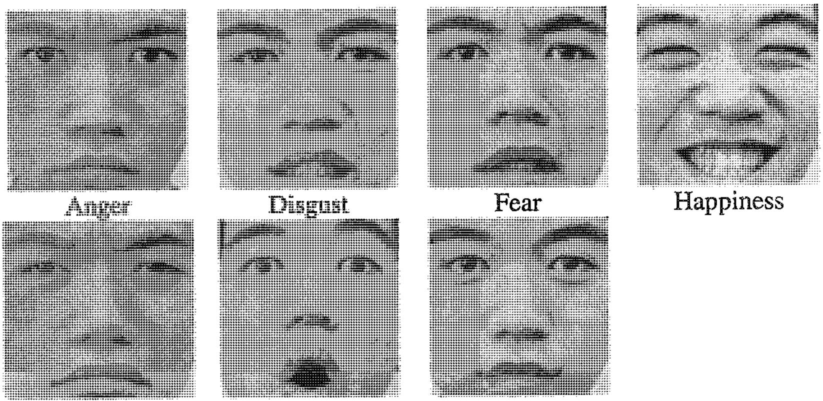
Next, we must convert these pictures into 2-bit encoded data in the frequency region to use as a training data. We use the 2-D DCT as a transformation technology because the face image variation is directly reflected to the frequency region and most of the energy concentrates on its low frequency part (Xiao *et al.* 1998).



Original image

Training image

Figure 2. A target image defined by three points.



Anger

Disgust

Fear

Happiness

Sadness

Surprise

Neutral

Figure 3. Samples of 6 emotional basic faces and neutral face.

2.3 Learning Experimental Results

In this chapter, we have used the pictures of the people faces and trained the sand glass type neural network. Each network learned some target images of one person to construct the emotional information in the third layer. When there are only fewer networks than the number of people, the network was not able to classify the people faces into each emotional face. Therefore, we used the sand glass type neural network which combined N neural networks. These networks have two neurons in the third layer and are interconnected. Figure 4 shows a typical neural network.

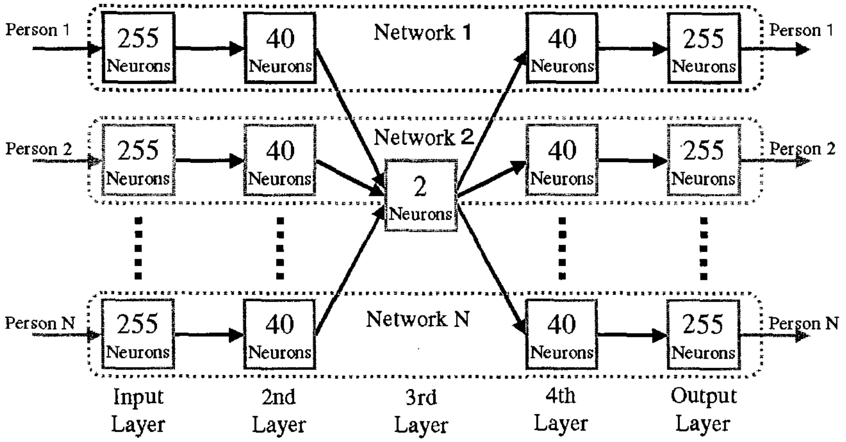
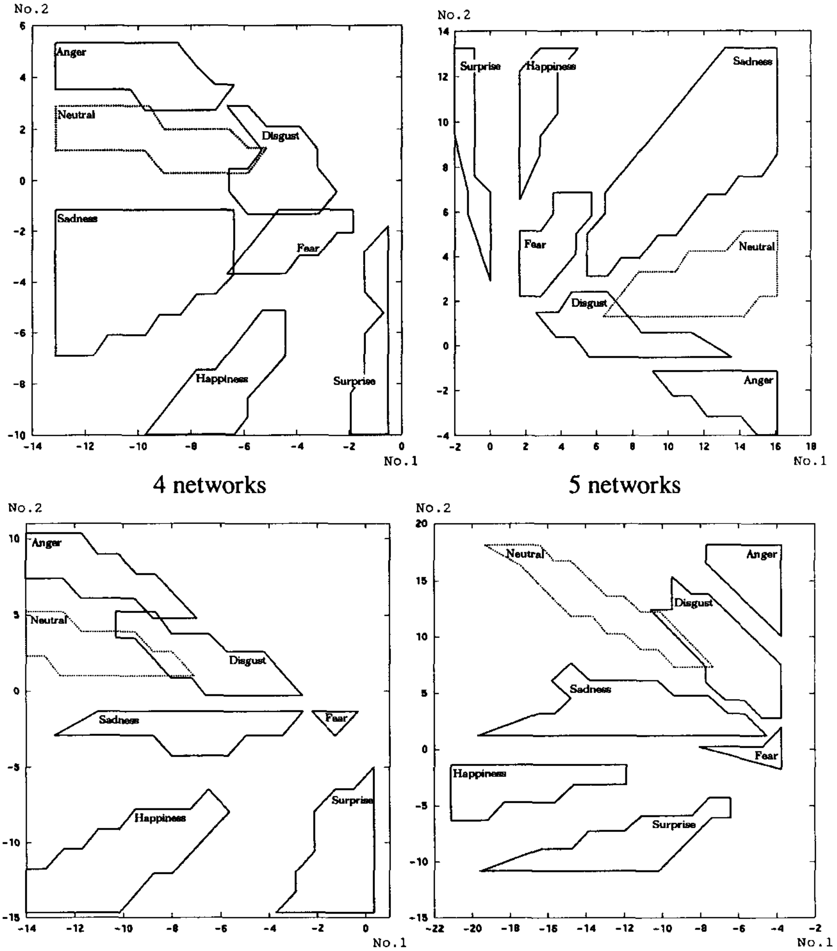


Figure 4. A typical neural network.

We consider that the information in output activities of two neurons is represented in the 2-dimensional emotional space. It is easy to understand this emotional space formed by the behaviors of neuron's output activities (Ichimura *et al.* 2001b). After training the network, we investigated the output activities in the third layer. Figure 5 shows the neuron output activities in the third layer. This figure is similar to the circumflex model of emotions by Russell and Bullock (1985) in Figure 6. However, it took long time to converge into the desired mean square error, because the network represents prominent characters of emotions by the output activities in the third layer.

2.4 Emotion Generating Calculations Method

Next, we explain a method for extracting emotions from the sentences. We consider 2 methods to generate the feeling. One method generates the emotions by analyzing the meaning of the sentences with personal taste information. Another method generates the mood from the summary of the extracted emotions. Emotion Generating Calculations (EGC) extracts pleasure/displeasure from the event represented in a sentence. This proposed method uses "favorable value" of the words in the event. Some favorable values are predefined and the others are obtained by knowledge acquisition. Under the extracted pleasure/displeasure by EGC method, their emotions are divided into 20 kinds of complex emotions based on Elliott's Emotion Eliciting Condition Theory (Elliott 1992).



4 networks

5 networks

6 networks

7 networks

Figure 5. Neuron activities in the third layer.

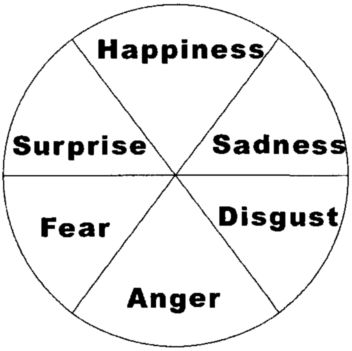


Figure 6. Emotion circle.

We can usually categorize human feelings into "pleasure," "sadness," "angry," "expectation," and so on. Such feelings are determined whether an event is pleasure/displeasure/unknown according to some words in the case frame representation. We define the equations of Emotion Generating Calculations according to the kinds of case frame representations as shown in Table 1. The f represents a favorable value of each type in case frame; f_S as "Subject," f_O as "Object," f_{OF} as "Object-From," f_{OT} as "Object-To," f_{OM} as "Object-Mutual," f_{OS} as "Object-Source," f_{OC} as "Object-Content," f_P as "Predicate." The 12 kinds of case frame structures are classified based on the following reasoning (Okada 1996):

- For Type I, Subject(S) does Verb(V) that influences reach to S.
- For Type II&III, S's statement which has a relation to V's changes from Object-From(OF) into Object-To(OT).
- For Type IV, S and Object-Mutual(OM) have a relation to V.
- For Type V, S and Object-Source(OS) do V at the same time.
- For Type VI-a), S does V to Object(O).
- For Type VI-b), O's statement is changed from OF into OT by S.
- For Type VII&VIII, O is done V by S.
- For Type X, O is done V using I by S.
- For Type XI, O has Object-Content (OC) as an attribute.
- For Type IX&XII, These types are fluctuated into various categories.

Table 1. Event type and Emotion Generating Calculation.

Type	Event type	Emotion Generating Calculation
I	$V(S)$	$f_S \times f_P$
II	$V(S, OF)$	$f_S \times (f_{OT} - f_{OF}) \times f_P$
III	$V(S, OT)$	$f_S \times (f_{OT} - f_{OF}) \times f_P$
IV	$V(S, OM)$	$f_S \times f_{OM} \times f_P$
V	$V(S, OS)$	$(f_S - f_{OS}) \times f_P$
VI	$V(S, O)$	a) $f_S \times (f_O \times f_P)$ b) $f_O \times f_P$
VII	$V(S, O, OF)$	$f_O \times (f_{OT} - f_{OF}) \times f_P$
VIII	$V(S, O, OT)$	$f_O \times (f_{OT} - f_{OF}) \times f_P$
IX	$V(S, O, OM)$	————————
X	$V(S, O, I)$	$f_O \times f_P$
XI	$V(S, O, OC)$	$f_O \times f_{OC}$
XII	Others	————————

Favorable value is the degree of feeling about like/dislike and it is a number in the range [-1.0, 1.0]. When this value is larger, it means liking more. And when it is smaller, it means disliking more. We decided the favorable value by the human's sense from the dialog's contents. We prepared the database related to favorable values where each emotional word has one value.

When for example the sentence "Romeo dates with Juliet" is given to the EGC method, the calculation is as follows:

Event: "Romeo dates with Juliet."
Predicate (P) = "dates with" : +0.5
Subject (S) = "Romeo" : +1.0
Object Mutual (OM) = "Juliet" : +0.9
Event type: "date with" • V(S, OM)
•
Emotion Value $= f_S$ (*Romeo*) $\times f_{OM}$ (*Juliet*) $\times f_P$ (*dates with*)
$= (+1.0) \times (+0.9) \times (+0.5)$
$= +0.45$ • positive number (pleasure)

The result shows that Romeo feels pleasure about the event "Romeo dates with Juliet."

2.5 Classification of Emotion Types

Based on such emotion values calculated by the EGC method and their situations, the degree of pleasure/displeasure for each emotion type is obtained. We consider only 20 emotion types, among 24 described by Elliott (1992). Figure 7 shows the dependency between the groups of emotion types. The 20 emotions are classified into some emotional groups as follows; "joy" and "distress" as a group of "Well-Being"; "happy-for," "gloating," "resentment," and "sorry-for" as a group of "Fortunes-of-Others"; "hope" and "fear" as a group of "Prospect-based"; "satisfaction," "relief," "fears-confirmed," and "disappointment" as a group of "Confirmation"; "pride," "admiration," "shame," and "disliking" as a group of "Attribution"; "liking"

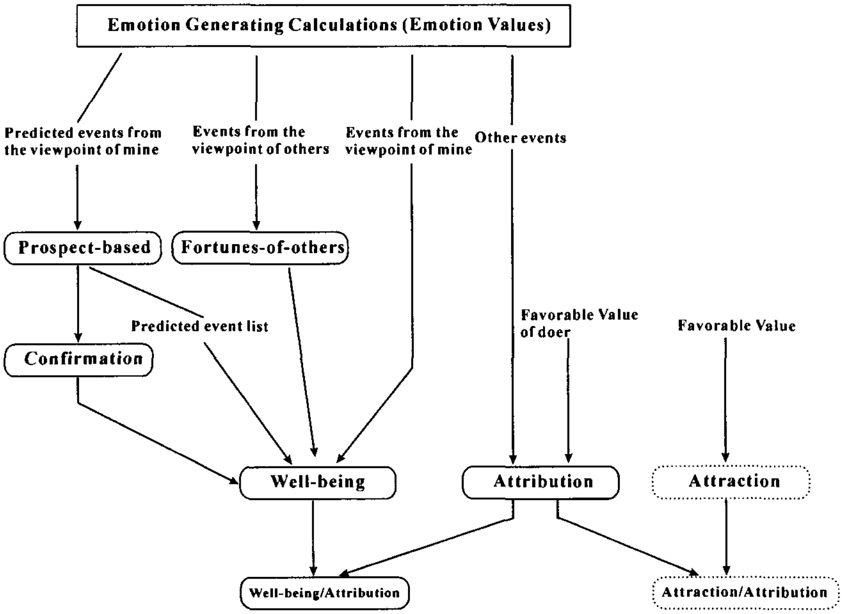


Figure 7. Dependency between emotion groups.

and "disliking" as a group of "Attraction"; "gratitude," "anger," "gratification," and "remorse" as a group of "Well-Being/Attribution"; "love" and "hate" as a group of "Attraction/Attribution." The emotion types, "liking" and "disliking," are not included in the generated emotion type by the EGC, since the emotion type "love" and "hate" are different from the others.

2.6 From E-Mail to Facial Expressions

The Facemail analyzes the sender's emotion type and degree from the e-mail content and represents a facial expression corresponding to the emotion of the person sending the e-mail. It needs to analyze the e-mail's content at the server side and interpret the determined emotional face in an e-mail header. The server has four stages of processing as shown in Figure 8; Morphological analysis and parsing, Translation into case frame format, Determination of facial expression, Message transfer with analysis results. As the server receives a message, it conducts a Morphological analysis and parsing, and translates it into case frame format. The EGC calculates the emotion type and degree corresponding to the e-mail. Although the EGC requires the favorable values and dictionary related to emotional words, our system has prepared the dictionaries on the basis of the questionnaire results for favorable values of some words.

We must consider how a facial expression is decided as a total face representation from the extracted emotions by the EGC. The 20 types of emotions by the EGC are translated into 6 kinds of facial emotion types by the assignment rules in Table 2. A parallel sand glass type neural network is trained for four selected characteristic facial pictures and the center point in an emotional space corresponding to the training data for each emotion is calculated as shown in Figure 9.

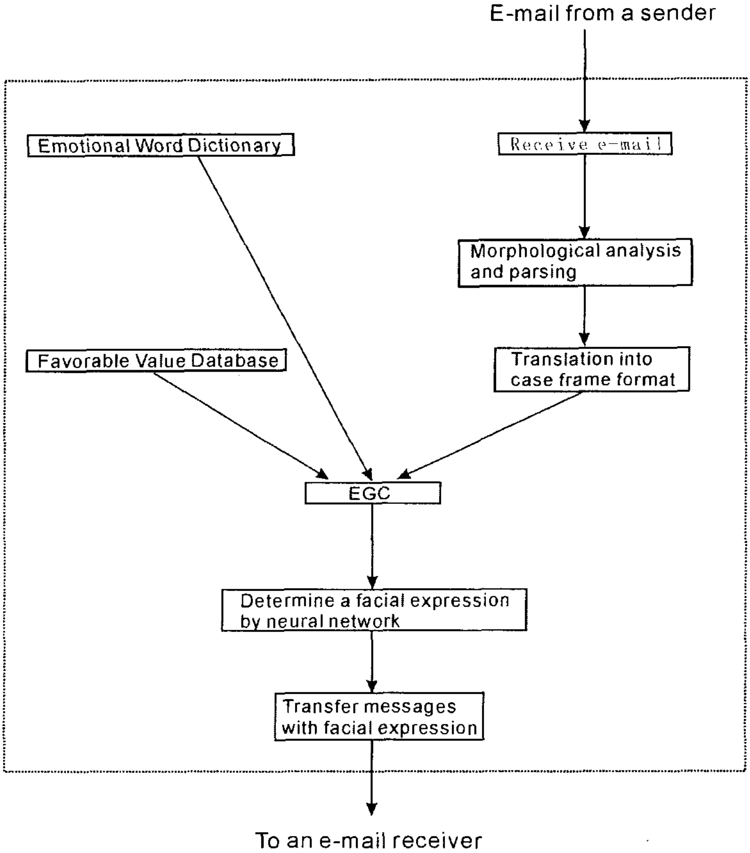

E-mail from a sender
Emotional Word Dictionary
Receive e-mail
Morphological analysis and parsing
Favorable Value Database
Translation into case frame format
EGC
Determine a facial expression by neural network
Transfer messages with facial expression
To an e-mail receiver

Figure 8. An overview of the system flow in the server.

Table 2. Assignment rules from EGC into facial emotion types.

Facial emotion type	Extracted emotions from EGC
Anger	anger
Pleasure	joy, happy-for, gloating, hope, satisfaction, relief, pride, admiration, gratitude, gratification
Displeasure	distress, resentment, sorry-for, fear, fears-confirmed, disappointment, shame, reproach, anger, remorse
Surprise	relief, disappointment
Disliking	none
Perplexity	sorry-for, fear, relief, disappointment

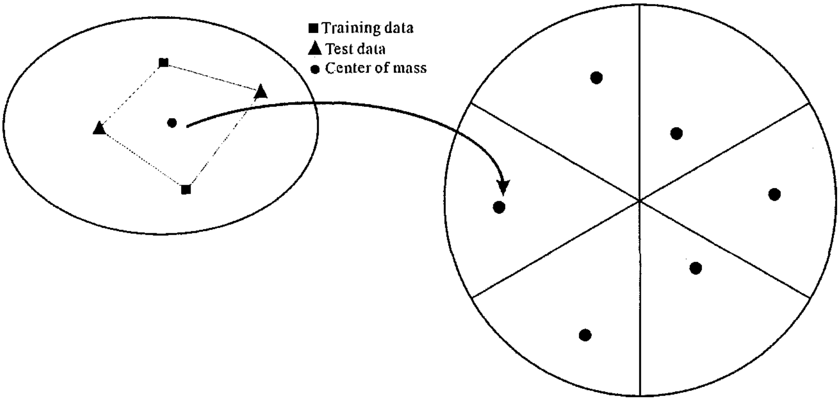


Figure 9. Input points in emotion space.

Furthermore, in order to centralize the 6 facial emotion types and their values we calculate a center vector of each vector as shown in Figure 10. Figure 11 shows the relation map between inputs to the emotional space and output activities in the fifth layer of neural network for a subject. The sentences in a mail are analyzed by proposed method, and then we can read the mail with a facial expression as shown in Figure 12.

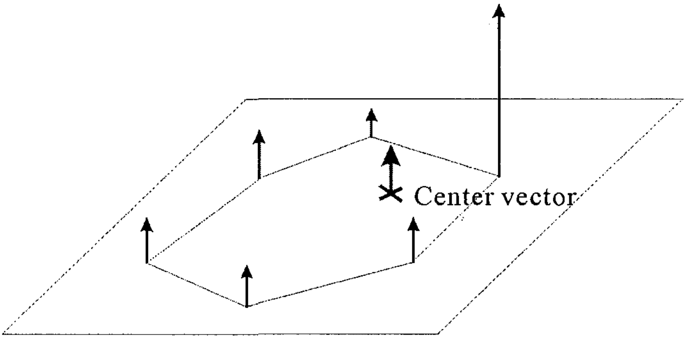


Figure 10. Center of 6 emotions.

However, in Facemail, the following problems still remain: it takes too long to train a parallel sand glass type neural network for a given facial expressions; in case of long e-mails, it shows a neutral face as a result of the summation of the various emotions in the different parts of the text. We will solve such problems in the near future, and then we will release a commercial-based human interaction system (Ichimura *et al.* 2001a,c).

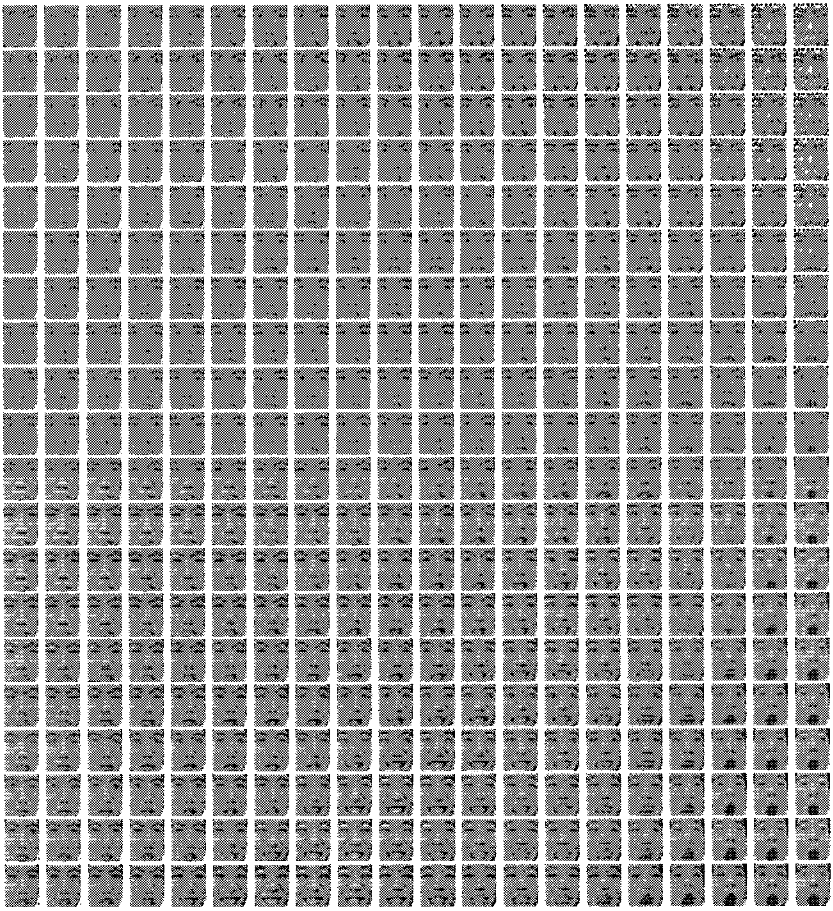
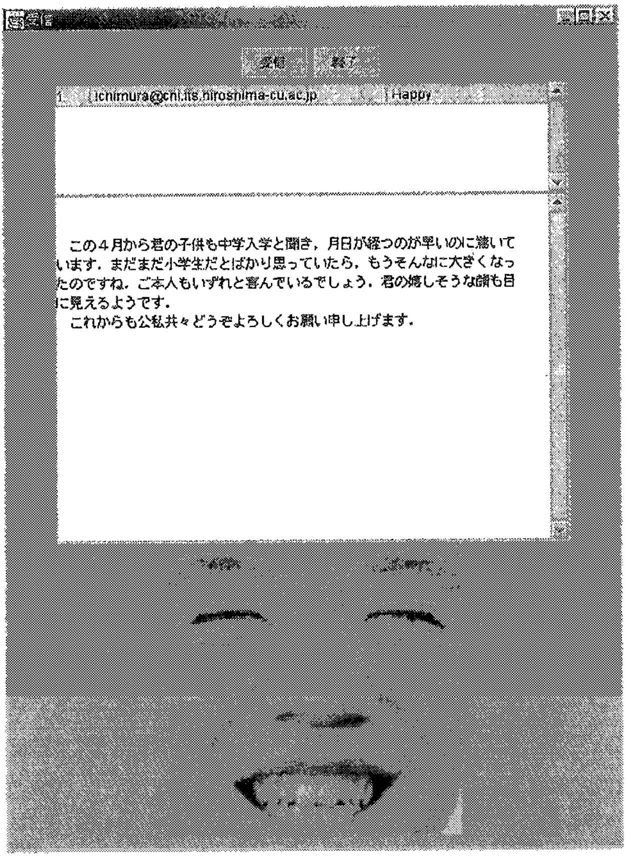


Figure 11. Relation map between emotional space and output activities.



1 ichimura@chi.its.hiroshima-cu.ac.jp Happy

この４月から君の子供も中学入学と聞き，月日が経つのが早いのに驚いています．まだまだ小学生だとばかり思っていたら，もうそんなに大きくなったのですね．ご本人もいずれと喜んでいるでしょう．君の嬉しそうな顔も目に見えるようです．

これからも公私共々どうぞよろしくお願い申し上げます．

Figure 12. Mail tool in Facemail.

3 Automatic Scenario Analysis System for Noh Play with Noh Masks

Noh play is one of the most popular traditional arts in Japan. Noh masks used in the Noh play are artificial and sometimes ambiguous, so they are interpreted as expressing varied emotions (Osaka 1986). In Noh masks, elements of the human face are combined and integrated in typical patterns, and their expressions are considered to summarize rich and diverse human emotions (Minoshita *et al.* 1999). On the stage, the Noh mask turns upward and downward or to the left and right, and the audience reads diverse emotions with the change of the angle of the mask. An upward turn of the mask, which is called *terasu* (shining), represents a pleasant and cheerful state of mind, and a downward turn of the mask, which is called *kumorasu* (clouding), represents a gloomy state of mind. The audience enjoys perceiving delicate emotions from the changes of the mask angle. In these days, the number of those who go to Noh theatre is decreasing because of the scarceness of Noh theatres. However, people can download the scenario for Noh play from the Internet and can read various scenarios for Noh play. But they cannot enjoy perceiving delicate emotions from the Noh mask.

For this reason, we propose a fuzzy reasoning model for selecting the image of Noh mask which expresses the emotions caused by some situation in the Noh play scenario. Moreover, we are constructing the system with which users can read the scenario of Noh play and can enjoy grasping delicate emotions from a Noh mask on the computer display.

We used *Koomote* as the Noh mask. *Koomote* is a type of female mask and is the most popular among Noh masks. *Koomote* is known to have the richest variety of facial expressions (Minoshita *et al.* 1997). We analyzed the subjects' responses to the relationships between the emotion items and the Noh mask images by MDS (multi-dimensional scaling) and identified the three dimensions: "attention – rejection," "pleasant – unpleasant," and "sleep – tension." For example, in Figure 13, the Noh mask images are plotted with the first dimension as the X-axis and the second dimension as the Y-axis. The first dimension is "rejection – attention," and the second dimension is "unpleasant – pleasant." In Figure 14, the Noh mask images are plotted with the first dimension as the X-axis and the third dimension as the Y-axis. The third dimension is "sleep – tension."

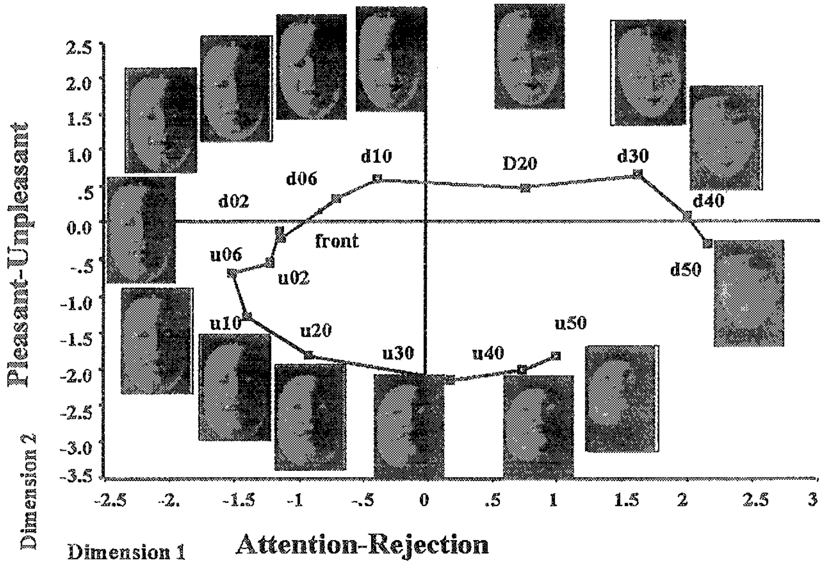


Figure 13. MDS plot of Noh mask image (Dimension 1, Dimension 2).

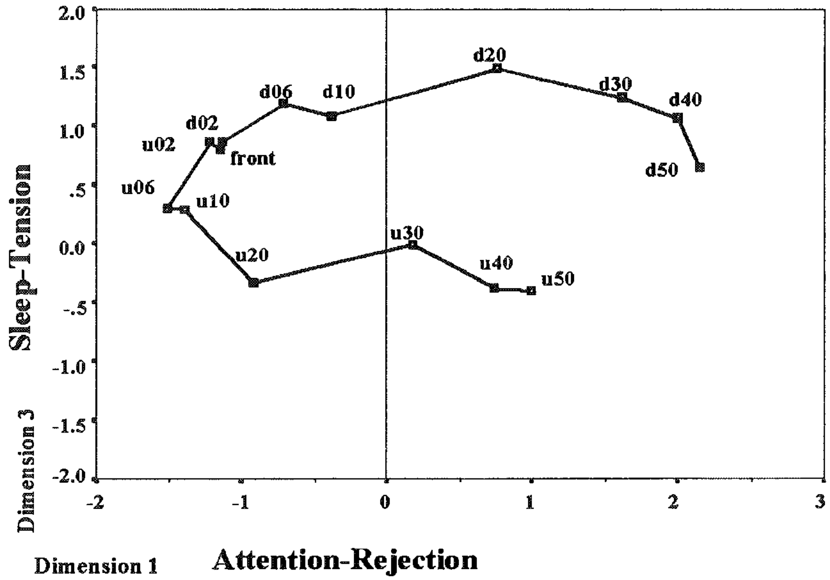


Figure 14. MDS plot of Noh mask image (Dimension 1, Dimension 3).

These MDS results agree with "pleasant – unpleasant," "attention – rejection," and "sleep – tension" described in the facial expression circle model presented by Schlosberg (1952), and accord with major dimensions of emotion perception obtained by dimensional evaluation methods for various expressions in the study of facial expressions.

It is surprising to note that the changing of the angle of a Noh mask can produce various emotions as well as human facial expressions.

3.1 Construction of Fuzzy Reasoning Model

We constructed the fuzzy reasoning model for selecting the angle of the mask which expressed the emotions caused by the sentences written in the Noh scenario as the following.

We used fifteen images of *koomote* turned from 50 degrees downward to 48 degrees upward (down50, down40, down30, down20, down10, down6, down2, front, up2, up6, up10, up20, up30, up40, up48), as shown in Figure 15.

As shown by Ekman and Friesen (1975), we used the following six basic emotions:

1. happiness
2. sadness
3. anger
4. disgust
5. surprise
6. fear.

We asked 69 undergraduates (63 males and 6 females) to choose one or several angles of the mask which expressed a given emotion. We assumed that the ratio of the subjects who selected the angle of the mask for a given emotion was the grade to which the facial expression of the mask belonged to a fuzzy set of facial expressions which expressed the emotion.

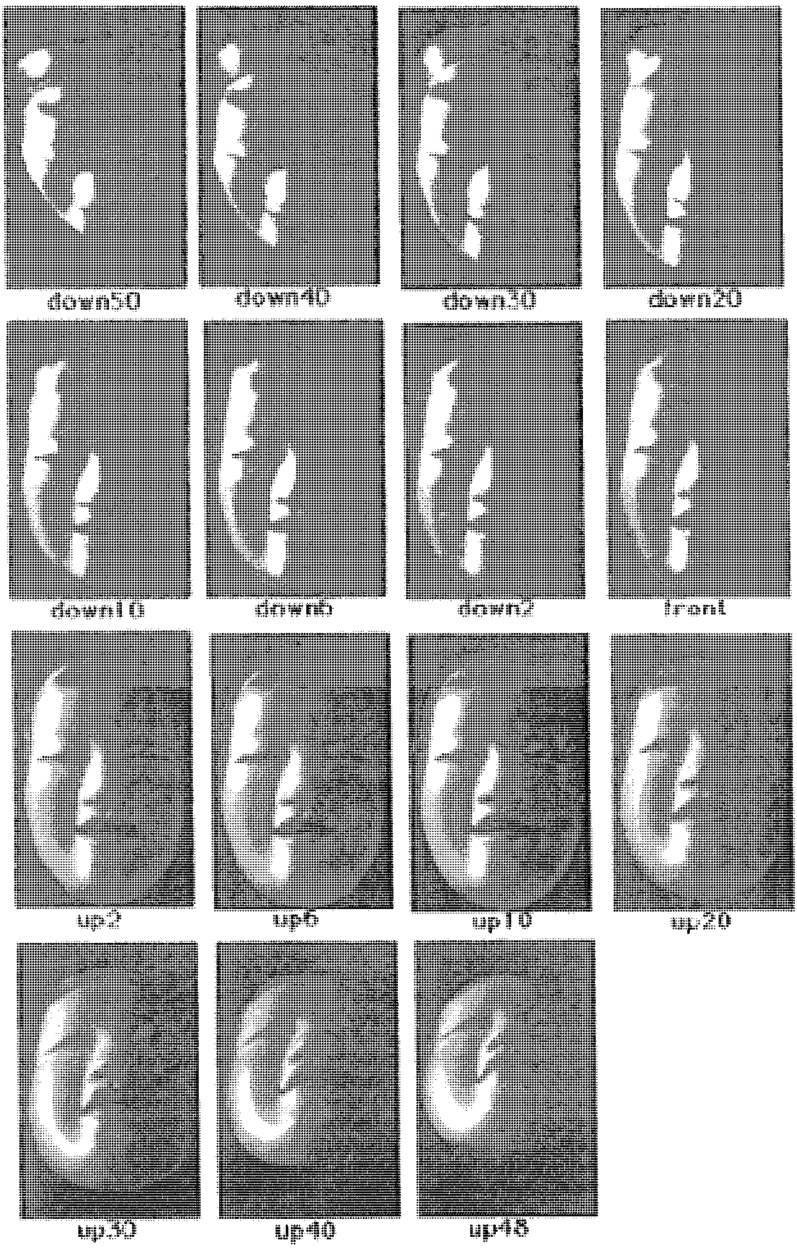


Figure 15. Noh mask images.

We proposed the following fuzzy reasoning model on the basis of Yamashita *et al.* (1999):

$$
\begin{array}{lllll}
\text{Rule 1:} & A_1 & & & \Rightarrow P_1 \\
\text{Rule 2:} & & A_2 & & \Rightarrow P_2 \\
 & & \cdots\cdots\cdots & & \\
\text{Rule 6:} & & & A_6 & \Rightarrow P_6 \\
\text{Input:} & a_1 \text{ and } a_2 \text{ and } \ldots \text{ and } a_6 & & & \\
\hline
\text{Conclusion:} & & & & P',
\end{array}
$$

where $A_1, A_2, \ldots$, and A_6 in the antecedent part are crisp sets of basic emotions, and $P_1, P_2, \ldots$, and P_6 in the consequent part are fuzzy sets of the facial expressions of the mask which express a given emotion.

If we have a_1, $a_2, \ldots$, and a_6 as the intensity of each emotion, then the fuzzy reasoning result of Rule i (i=1, 2,...,6) is calculated by

$$\mu_{Pi'}(z) = a_i \wedge \mu_{Pi}(z),$$

and the combined conclusion of Rule 1, Rule 2, ..., and Rule 6 is given as

$$\mu_{P'}(z) = \mu_{P1'}(z) \vee \mu_{P2'}(z) \vee \cdots \vee \mu_{P6'}(z).$$

The angle of the mask with the highest or the second highest grade of membership should be selected as the angle of the mask which well expresses the emotions caused by a given situation.

3.2 Application of the Model

We are applying our fuzzy reasoning model to the automatic scenario analysis system which selects and displays the Noh mask image with the scenario for Noh play.

First, our system receives the sentences in the scenario. Then the system decomposes the sentences into words and transforms these words into original forms in terms of inflection. These words are compared to the words in the database for each emotion. Then, the membership values for all the emotions are calculated. These membership values are used as the input values for fuzzy reasoning or neural network. As a result of fuzzy reasoning or neural network, an image of the Noh mask with the maximum membership value is selected and displayed.

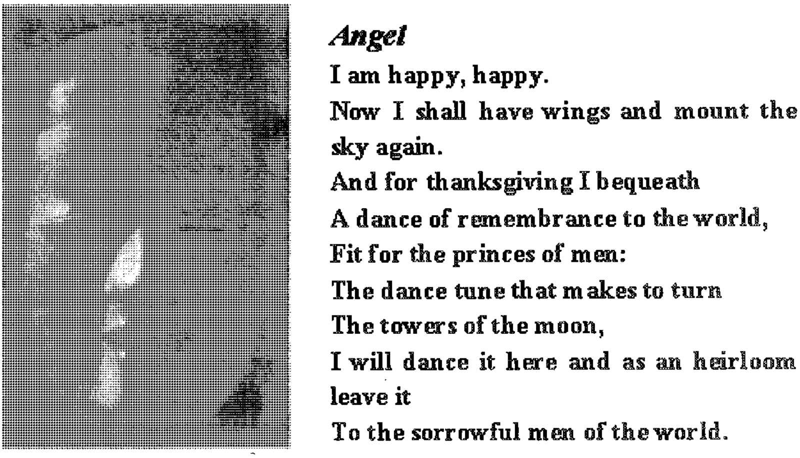
For example, the sentences in the scenario (Waley 1997) are:

I am happy, happy.
Now I shall have wings and mount the sky again.
And for thanksgiving I bequeath
A dance of remembrance to the world,
Fit for the princes of men:
The dance tune that makes to turn
The towers of the moon,
I will dance it here and as an heirloom leave it
To the sorrowful men of the world.

As for these sentences, the first and second underlined word “*happy*,” and the third underlined word “*thanksgiving*” are identified as representing the emotion “happiness.” The fourth underlined word “*sorrowful*” is identified as “sadness.” Therefore, “sadness” has a membership value of 0.25 (=1/4) and “happiness” has a membership value of 0.75 (=3/4). These membership values are used as the input values for fuzzy reasoning.

As a result of fuzzy reasoning, the image “down10” with the highest membership value is selected and displayed as shown in Figure 15.

We used “*Hagoromo*” as a Noh play. The story of “*Hagoromo*” is as follows: a fisherman finds an angel’s cloak on the beach. He steals it and so prevents her return to heaven. The angel is very sad because she cannot go back to the sky. After several conversations, the fisherman decides to give the cloak back to the angel. The angel is very glad and dances showing the glory of heaven. Figure 16 shows the scene where the angel is very glad to know that she will be able to go back to heaven.



Angel
I am happy, happy.
Now I shall have wings and mount the sky again.
And for thanksgiving I bequeath
A dance of remembrance to the world,
Fit for the princes of men:
The dance tune that makes to turn
The towers of the moon,
I will dance it here and as an heirloom leave it
To the sorrowful men of the world.

Figure 16. An example of display.

4 Impression-Based Retrieval System of Music Works

One of the media data main research topics is how to retrieve a data from a database. Many research projects have proposed content-based retrieval methods for music works using a note sequence which is inputted by singing or using a musical keyboard (Tseng 1999, Rolland *et al.* 1999, Melucci and Orio 1999). These methods are effective when the user knows some information about the music work.

Recently, a computer with a network device has high functionality and is low in price. Therefore, we can easily create an original media data such as a music work, an image, or a video. Moreover we can spread the original media data or can collect different kinds of media data through the network. Then the end users can easily create the multimedia contents from the network-collected data without creating each media data composing it.

When we consider the above case of creating multimedia contents, the multimedia contents creator should often need a media data that is close to his/her impression. A typical example is when a creator should select the most suitable work as background music for a corresponding image from a group of music works that he/she does not know at all. The previously mentioned methods cannot be used if a user does not have some information about the required music work.

4.1 The Affective Value of Music

In the field of the psychology of music, there are some researches measuring the emotion of subjects by listening to music works. Hevner (1936, 1937) investigated the relations between the music structure and the emotion response by using the adjective check-list. Taniguchi (1995) has shown the relationship between the affective value, which is a certain degree of our impression in the emotional side of a music work, and the emotion response obtained by listening to them.

Taniguchi (1995) defined the affective value scale of music (AVSM) as a scale to measure the affective value of music works. In his method, the affective value corresponded to the quality and the intensity of a given affect. AVSM has five major factors and 24 adjectives as shown in Table 3. In order to measure the affective value by AVSM, the subjects allocate the 24 adjectives into an affective sound in a music work on a five-point scale: (1) not entirely fit, (2) not fit, (3) neither, (4) fit, and (5) just fit.

The affective value consists of five scores, one for each factor. The score of a factor is calculated as a summation of the five-scale points for the adjectives included in that factor. However, the "lift" factor consists of negative adjectives and positive ones. Therefore, we divide the sum of the two points in the lift factor by 2 and calculate the average point. Thus, the maximum value of the points in each major factor is 20, the minimum value is 4, and the median point is 12. Furthermore, Taniguchi reported the affective values for 90 pieces of classical music works by this measurement.

Table 3. Five major factors and twenty-four adjectives of AVSM.

Factor	Adjectives			
Lift (positive)	Cheerful	Joyful	Happy	Bright
Lift (negative)	Melancholy	Pathetic	Mournful	Gloomy
Affection	Tender	Longing	Sweet	Tranquil
Strength	Robust	Intense	Exciting	Emphatic
Frivolousness	Whimsical	Merry	Light	Restless
Solemnity	Solemn	Awe-inspiring	Lofty	Dignified

4.2 An Overview of the Proposed System

We develop an impression-based retrieval system for music works by impression words. The outstanding characteristic of our proposed system is the application of the AVSM method to a database. Figure 17 shows an overview of the proposed impression based retrieval system. An affective value of music works measured by AVSM consists of 5 points where each point is selected in 5-point scale for each major factor. Therefore, its affective value is a 5 dimensional vector called an affective vector of music works and each vector element is the selected point for the 5 major factors.

In this system, the user inputs an affective vector of a desired music work as a query. The result is a music work from the database whose affective vector is closest to the affective vector of the query. In order to find the closest affective vector, we measure the Euclidean distance between the affective vector of the query and each affective vector stored in the database. When we use an affective vector as a query, we can use the same method as the one used to measure an affective value by AVSM. In other words, a user allocates the 24 adjectives into affective sound in a desired music work. Our method adopts the 19 affective values of music work reported by Taniguchi (1995).

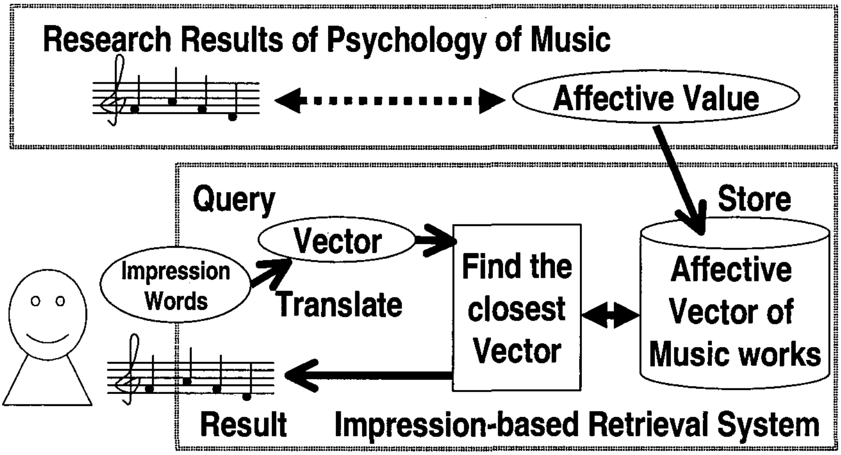

Research Results of Psychology of Music
Affective Value
Query
Store
Vector
Impression Words
Translate
Find the closest Vector
Affective Vector of Music works
Result
Impression-based Retrieval System

Figure 17. Overview of the proposed impression-based retrieval system.

4.3 Problem of Our Proposed Retrieval System

In order to make our proposed retrieval system practical we may need a lot of pairs of media data and its corresponding affective value in order to build the database. The measurement of the affective value by AVSM requires experiments in which many subjects listen to the music work. The building cost of such a database is very high. Therefore we propose a new method which can build a database with a lower cost.

Our proposed method is to calculate the affective values based on the results of the music structure analysis. Since the affective value has an aspect of human emotion while people listen to a music work, we think that the framework of the same music work influences the affective values. If the relation between the affective values and a structure of music framework is represented explicitly, the affective values can be calculated automatically from a music work structure. Most of the current music work is stored in digital representation such as MIDI (Musical Instruments Digital Interface) format, from which it is easy to analyze the music structure.

The analysis of music works is shown in Section 4.4. We propose two estimation methods of the affective values by using MIDI music works. The first one shown in Section 4.5 is the search method for structure of music works applying the technology of the machine learning. The second one shown in Section 4.6 is the search method using fuzzy regression analysis.

4.4 Analyzing Method of Music Structure Using MIDI Format Data

A MIDI format is being used as a data format for transfer between the MIDI instruments. We can directly listen to the music data in MIDI format using MIDI instrument. Moreover, each computer is equipped with MIDI instrument in its soundcard. It is easy to collect the MIDI data from the Internet as well as to analyze it on computer because it is an open standard format.

Hevner (1936, 1937) has reported the effect of 6 music structures (tempo, pitch, mode, melodic line, rhythm and harmony). We analyzed the degree of agreement values in the following 6 music structures:

- **Tempo**: 3 categories; slow, medium, or fast tempo
- **Mode**: 2 categories; major or minor mode
- **Melodic line**: 3 categories; rising, flat, or falling melodic line
- **Rhythm**: 2 categories; medium, or flowing rhythm
- **Harmony**: 2 categories; simple or complex harmony
- **Meter (music times)**: 3 categories; simple, compound, or peculiar times.

Based on the analysis results of Yoshino *et al.* (1998), we obtained the feature values in the interval [0,1]. Each feature value represents the rate of the frequency with which a music structure occurs. We used 37 classical music works in MIDI format, downloaded from the Internet, which were also among the 90 music works used by Taniguchi (1995).

4.5 Estimation Method by Machine Learning

In order to examine the relationship between the affective value and the feature value, we use the C4.5 Quinlan's method (Quinlan 1993). This is a popular method to discover and analyze some patterns included implicitly in the classification of numerous records. The method also generates a classifier in the form of a decision tree from which a set of production rules are derived.

In Quinlan's method, each recode is regarded as a case. A case consists of some attributes and a predefined class. In this section, we use the feature values as such attributes and assign the 5 major factors of the affective values into 5 predefined classes based on the arithmetic mean and the standard derivation as shown in Table 4. The table shows the result of the questionnaires on the affective values. Then, we obtained the production rules with respect to the relation between the affective values and the feature vector by C4.5 method as shown in Figure 18.

Table 4. Statistics of 37 affective values.

Factor	Arithmetic mean	Standard deviation
Lift	12.40	3.56
Affection	10.96	2.83
Strength	9.07	3.38
Frivolousness	8.45	2.20
Solemnity	10.04	1.34

4.6 Estimation Method by Fuzzy Regression Analysis

In this section, we analyze the relation between the affective values as statistical value of psychological experimental results and the feature vector as the structure of music works by fuzzy linear regression. We obtain some parameters of a fuzzy linear model based on the solution of fuzzy regression. Since the proposed model can estimate the affective value, the affective value is determined in a lower cost.

1. lift

Rule 1:	Tempo(Slow)>0.003977 Tempo(Fast)>0	⇒ class A
Rule 2:	Tempo(Slow)≤0.003977 Tempo(Fast)>0	⇒ class D
Default:	class C	

2. affection

Rule 1:	Rhythm(Flowing)≤0.285714 Melody(Flat)>0.319412	⇒ class E
Default:	class C	

3. strength

Rule 1:	Tempo(Fast)>0.32998 Meter(Compound)>0.009714	⇒ class C
Rule 2:	Tempo(Fast)≤0.32998 Meter(Compound)>0.009714 Rhythm(Flowing)≤0.513514	⇒ class E
Rule 3:	Tempo(Midium)≤0.257828 Rhythm(Median)>0.659514	⇒ class A
Rule 4:	Tempo(Fast)≤0 Meter(Compound)≤0.009714 Rhythm(Median)≤0.659514	⇒ class D
Rule 5:	Rhythun(Flowing)>0.513514	⇒ class D
Default:	class A	

4. frivolousness

Rule 1:	Meter(Simple)>0 Meter(Simple)≤0.990286	⇒ class B
Rule 2:	Rhythm(Medium)≤0.739583	⇒ class C
Rule 3:	Harmony(Simple)>0.67759 Rhythm(Medium)≤0.739583 Melody(Raising)>0.241918	⇒ class D
Rule 4:	Meter(Simple)≤0	⇒ class D
Default:	class C	

5. solemnity

Rule 1:	Key(Minor)>0.727701	⇒ class B
Rule 2:	Tempo(Fast)>0.989712 Tempo(Fast)≤0.994333	⇒ class B
Rule 3:	Tempo(Medium)>0.03326 Tempo(Medium)≤0.257828	⇒ class D
Rule 4:	Tempo(Medium)>0.257828	⇒ class C
Rule 5:	Meter(Mixed)>0	⇒ class C
Default:	class C	

Figure 18. Calculating production rules for classifying feature values into affective values.

In this method, under the condition of a pair of mean and variance in an affective value, we assume that a fuzzy number in the space of the affective values can be represented as the fuzzy linear model for a given feature vector. Using the fuzzy linear regression, we analyze the combinations of the statistical value of the psychological experimental results and the feature vector as the structure of music works. From this solution of regression, we get the parameters of the fuzzy linear model. As the affective value can be estimated by using this model, the cost to get the affective value can be made lower.

Tanaka *et al.* (1987, 1988, 1991) have formalized several models of fuzzy regression analysis when an input-output pattern ($\boldsymbol{x}_i$, $\boldsymbol{Y}_i$) satisfies with a given fuzzy linear system. In this formulation, fuzzy numbers are symmetric fuzzy numbers. A membership function of symmetric fuzzy number A_i denoted as (α_i, c_i) is defined

$$\mu_{A_i}(a_i) = L((a_i - \alpha_i)/c_i)$$

where a reference function $L(x)$ satisfies with (i) $L(x) = L(-x)$, (ii) $L(0) = 1$ and (iii) L is strictly decreasing function in $[0, +\infty]$.

The affective value is represented as a symmetric fuzzy number $A = (\alpha, c)_L$ where α is mean and c is variance. $L(x)$ is defined by the following equation:

$$L(x) = e^{-x^2}$$

The input-output pattern is some combination of the affective value "frivolousness" (Y_i) and the feature vector ($x_{i1}, \ldots, x_{i15}$). The fuzzy linear model is $Y_i = A_0+A_1x_{i1}+\ldots+A_{15}x_{i15}$. We can treat it as a minimize problem and an equation $L(x)$ is considered as shown in Table 5. From this solution, a raising melodic line had effects on "frivolousness" positively and a flat melodic line had effects negatively.

Table 5. A solution of fuzzy linear regression.

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
α	11.04	0.52	0.00	–0.60	–1.41	0.00	–2.40	0.00
c	2.45	0.00	0.22	0.83	0.00	0.59	0.00	0.14
Index	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
α	1.27	2.26	0.00	–0.45	0.00	1.96	–1.46	0.00
c	1.02	0.52	0.00	0.03	0.00	0.78	0.00	0.25

4.7 Hybrid Method of Machine Learning and Fuzzy Regression Model

We proposed two estimating methods of the affective values and feature values; applying the C4.5 method, a popular technique in machine learning and using the fuzzy regression analysis.

The first method is effective in a rough estimation because it is difficult to estimate the detailed output from the calculation result. On the other hand, the second method is excellent in human feeling estimation, but an error in the model arises from the fuzzy regression analysis when the measurement contains an extreme value. In order to improve the estimated precision, we need to develop a new estimation method where the advantage of the two methods is extended by combining the both. That is, the relation between the affective values and the feature values are classified into some categories by the first method and the model in each category is built by fuzzy regression analysis.

We must note that the precision of the two methods is different for each major affective factor. However, we consider that the classification of the first method is more detailed if generally the precision of the second method is not good.

The number of the measurements has an effect to the estimation precision of the both methods. To improve the precision of our methods, we have to obtain a lot of measurement data. For example, in order to use all affective values of the 90 classical music works reported by Taniguchi (1995), the feature vector is extracted from the sound data. The development of this extraction method is left as a future research subject.

5 An Artificial Emotion Processing System and Its Color Expression

The study of human-friendly model, which can help to design a computer that deals with human sensibility, is important. However, this is a difficult problem due to the fact that it includes subjectivity, vagueness, ambiguity and conditional dependence. In particular taking subjectivity into account, we proposed the theory for Subjective Observation Model (Shirahama and Miyamoto 2001). There are many interesting applications of this theory and we have also implemented few emotional processing systems (Shirahama 2000, 2001).

To construct an emotional database has long been a major focus of our research. The emotional data information is normally obtained through a questionnaire, as it remains the most popular method to build an emotional database. However, it is difficult to apply questionnaire methods to construct a general emotional database. The obtained information through the questionnaire depends on the questionnaire entries, the date or its group. In this section we have showed that the emotional database comes not only from a questionnaire but also from theories of emotional psychology. The emotional database described here is very simple, but it is important in order to construct the standard data of emotions. We call "formal emotional database", the database based on the theories of emotional psychology. We have constructed an artificial emotional model based on the formal emotional database.

5.1 Vector Expression of Emotions

In this section, we explore the concept of artificial emotions based on the theory for Subjective Observation Model. We assume an artificial emotion model, which is expressed as multi-dimensional vector in Euclidean space. We call "image code" a vector including emotional data and we define a set of emotional image codes as emotional image code database.

The emotional psychology treated here is based on the theories of Plutchik (1989). He proposed a multidimensional model of emotions where the constructive idea came from the analogical inference for the three dimensional mixed color model (Kawata 1976). The following six postulates are proposed to construct the model:

Postulate 1. There are a small number of pure or primary emotions.

Postulate 2. All other emotions are mixed, that is, they can be synthesized by various combinations of primary emotions.

Postulate 3. Primary emotions differ from each other, both with regard to physiology and behavior.

Postulate 4. The emotions of daily life are mixed.
Postulate 5. Primary emotions can be conceptualized in terms of a pair of polar opposites.
Postulate 6. Each emotion can exist in various degrees of intensity.

Under these 6 postulates, 4 attributes are introduced corresponding to the 4 pure pairs of emotional words [joy-sadness], [anger-fear], [expectation-surprise], [acceptance-hatred]. These attributes are expressed later by [JOY.], [ANG.], [EXP.] and [ACC.]. Figure 19 shows the structure of the 4 pure emotion pairs.

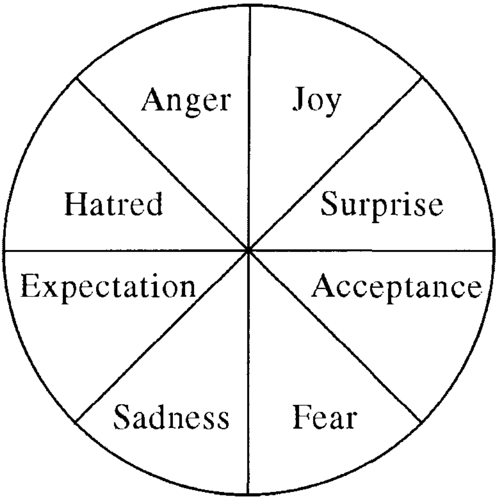


Figure 19. A part of formal emotion Database (DB) and adjusted emotion DB.

According to the second postulate, the mixed emotions can be defined on the 4 attributes and expressed by the 4 dimensional vectors. Table 6 shows 24 mixed emotions defined by various combinations of pure emotions of Figure 19.

The weights of the attributes for each emotion are done within the range [-1, +1]. Table 7 shows the emotional image code of 8 pure emotions and the part of emotional image code of mixed emotions.

We define that the weights of the attributes of each mixed emotion are 0.71 or –0.71 considering that all emotions are on 4-dimensional hyper sphere.

Table 6. 24 mixed emotions based on each pair of emotions.

Primary Pair	Secondary Pair	Tertiary Pair
joy + anger *= pride*	*joy + expectation* *= optimism*	*joy + acceptance* *= affection*
anger + expectation *= attack*	*anger + acceptance* *= superiority*	*anger + sadness* *= indignation*
expectation + acceptance *= admission*	*expectation + sadness* *= pessimism*	*expectation + fear* *= uneasiness*
acceptance + sadness *= sentimentality*	*acceptance + fear* *= obedience*	*acceptance + surprise* *= curiosity*
sadness + fear *= despair*	*sadness + surprise* *= disappointment*	*sadness + hatred* *= regret*
fear + surprise *= astonishment*	*fear + hatred* *= disgrace*	*fear + joy* *= guilt*
surprise + hatred *= contempt*	*surprise + joy* *= delight*	*surprise + anger* *= abhorrence*
hatred + joy *= unhealthy*	*hatred + anger* *= resentment*	*hatred + expectation* *= irony*

According to Postulate 6, we introduce 3 degrees of intensity of emotions. For example, strong emotions are set on 4-dimensional hyper sphere surface (see Figure 20). Thus 3 degrees of intensity are introduced to 32 emotions (8 pure emotions and 24 mixed emotions). Thereby we build a formal emotional image code database which includes 96 emotions.

Table 7. Emotional image codes.

Formal Emotion DB

Emotional word	JOY.	ANG.	EXP.	ACC.
Joy	0.67	0.00	0.00	0.00
Anger	0.00	0.67	0.00	0.00
Expectation	0.00	0.00	0.67	0.00
Acceptance	0.00	0.00	0.00	0.67
Sadness	–0.67	0.00	0.00	0.00
Fear	0.00	–0.67	0.00	0.00
Surprise	0.00	0.00	–0.67	0.00
Hatred	0.00	0.00	0.00	–0.67

Adjusted Emotion DB

Emotional word	JOY.	ANG.	EXP.	ACC.
Joy	0.75	0.00	0.50	0.30
Anger	0.00	0.75	–0.50	0.00
Expectation	0.20	0.00	0.65	0.30
Acceptance	0.00	0.00	0.40	0.35
Sadness	–0.70	0.00	–0.40	–0.40
Fear	–0.10	–0.70	–0.30	–0.30
Surprise	0.00	0.00	–0.65	0.00
Hatred	0.00	0.30	0.00	–0.70

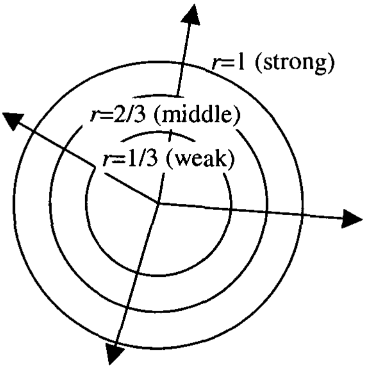


Figure 20. 3 degrees of intensity of emotions.

5.2 The Theory for Subjective Observation Model

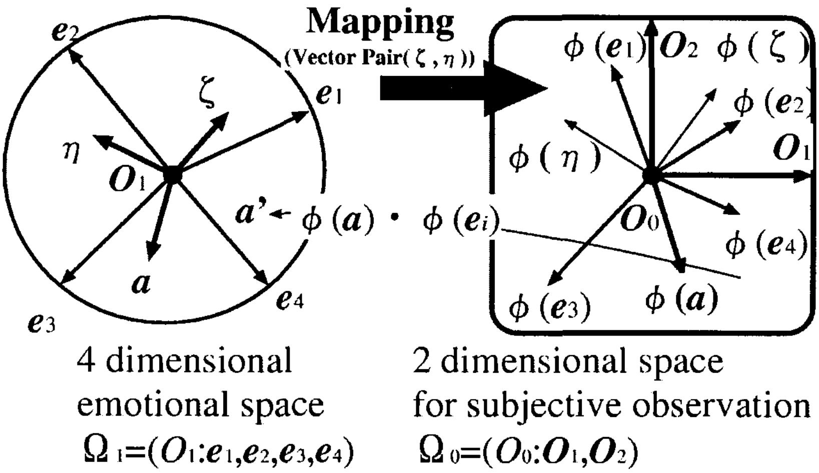
The subjective observation model is presented in this section. Image codes of the mixed emotions $\{x_k\}$ (k=1, 2, ..., 96) can be expressed by the vectors consisting of the coefficients on four attributes of a Normalized Rectangular Basal Coordinate (NRBC) whose system is denoted by

$$x_k = x_{k1}e_1 + x_{k2}e_2 + x_{k3}e_3 + x_{k4}e_4,$$

where

$$(e_1,e_2,e_3,e_4) = (\text{[JOY], [ANG.], [EXP.], [ACC.]}).$$

Human beings are in general almost unable to directly understand the absolute meanings of the objects or the relationships among the objects in more than a 3-dimensional space. Usually, we recognize and understand the meanings concerning the objects by dropping the order of dimensions and aggregating the observed information from several angles.



4 dimensional emotional space $\Omega_1=(O_1:e_1,e_2,e_3,e_4)$

2 dimensional space for subjective observation $\Omega_0=(O_0:O_1,O_2)$

Figure 21. Image diagram of subjective observation.

A specialized 2-dimensional Euclidean space for observation is developed. The model stands on the philosophical point that all recognition and understanding can be done by mapping the objects defined in a higher dimensional space onto the observational space (See Figure 21). The observational space dimension is not necessarily two, however we have determined it due to a convenience of the observation.

5.3 Computer Simulation

In order to clear the differences between formal emotional image code database and old emotional image code database by pair vectors (obtained by our questionnaire), the explanation using the maps onto 2-dimensional plane is introduced here. First, we select 2 vector pairs (ζ, η), each of which is constructed by the 2 emotional words, such as [abhorrence, attack] or [optimism, curiosity]. The vector pairs are used for the mapping function, ϕ_{ABH_ATT}. We consider that the mapping function ϕ_{ABH_ATT} corresponds to a kind of person's character for which the point of view is abhorrence and attack. The mapping function ϕ_{ABH_ATT} maps any emotional image code vector $\boldsymbol{x}_k$ in 4-dimensional emotional space onto the 2-dimensional observational space. The distribution of the mapped points $\phi_{ABH\_ATT}(\boldsymbol{x}_k)$ is shown in Figure 22. All emotions are drawn by red point. Then we get the pseudo-vectors $\boldsymbol{x}'_k$ for the subjective observation applying the mapping function ϕ_{ABH_ATT}. Finally, as for any point on the observation space, it can be categorized to some emotional words as well as the mapped point $\phi(\boldsymbol{x}_k)$. Considering the point $\boldsymbol{P}_m(p_{1m}, p_{2m})$ on the observation space and performing the inverse mapping operation, we can obtain the corresponding image code in 4-dimensional space, and can find the emotional word closest to where it is located. In more mathematical details, by computing the equation

$$q_{im} = p_{1m}\zeta_i + p_{2m}\eta_i, \text{ (for } i = 1, 2, 3, 4\text{)},$$

we can obtain the vector $Q_m(q_{1m}, q_{2m}, q_{3m}, q_{4m})$ and find the emotional words.

In this way, we can obtain the map of the categorized emotion and draw a clustering map of emotional space into regions of emotional classes as shown in Figure 22 and Table 8. Each attribute of the emotional image code has one to one correspondence to RGB. Therefore, we can see the maps colorfully. The 4 attributes of the emotional space, [JOY], [ANG], [EXP] and [ACC] correspond to red, green, blue and brightness, respectively. It is convenient to use a color expression to visualize an emotional distribution.

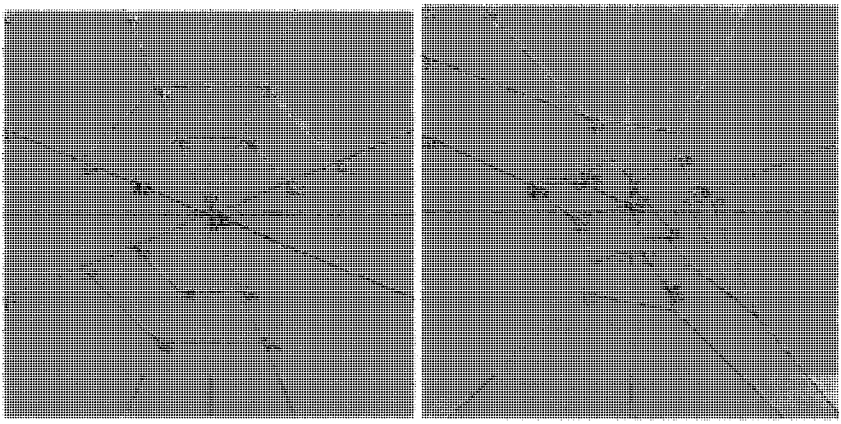


Figure 22. [Abhorrence, attack].

Table 8. [Abhorrence, attack].

Emotion Area	No.	Emotion Area	No.
weak hatred	24	anticipation	8
weak anger	15	abhorrence	6
weak expectation	15	posture	6
weak fear	15	contempt	5
weak surprise	15	detest	4
weak attack	3	boredom	4
weak astonishment	3	awe	4
weak uneasiness	3	astonishment	4
weak abhorrence	3	obedience	4
		obstinate	3
		surprise	3

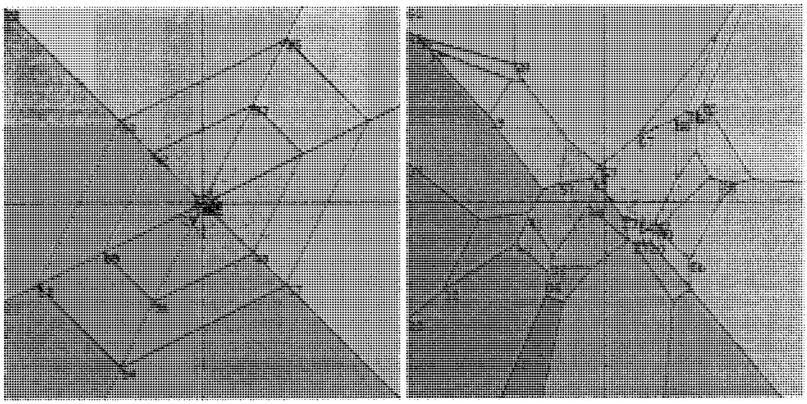


Figure 23. [Optimism, curiosity].

Table 9. [Optimism, curiosity].

Emotion Area	No.	Emotion Area	No.
weak curiosity	23	calm	18
weak irony	23	trouble	13
weak optimism	17	meditation	12
weak disappointment	17	surprise	7
weak hatred	6	pleasure	6
weak affection	3	irony	3
weak regret	3	optimism	2
Optimism	1	envy	2
disappointment	1	disappointment	2
Curiosity	1	posture	1
Irony	1	modesty	1
		curiosity	1

5.4 Future Work

We have presented our artificial emotion model implementing a formal emotional image code database. This artificial emotion model has subjective processing system based on mapping function. From the diagram of the mapped emotional 2-dimensional plane and its color expression it is possible to see the mapping function-oriented clustering algorithm. We defined the mapping function as an operator that interprets image codes, which are expressed by emotional words and vectors. As this is a novel study, a lot of issues remain for further research. Therefore, in the near future, we might need to create some new processing system, including emotional functions.

6 Conclusion

In this chapter, we have described an emotion oriented intelligent system which enables human-like communication. The system has various functions: to realize nonverbal messages such as facial expressions, to analyze sentences in letter or scenario, to quantize an emotion when we listen to music or watch pictures and so on.

We developed the technique for understanding human's emotion or to embed the emotion's function into a computer. Our research field of engineering aims to achieve a smooth communication between humans and the computers. However, it is too difficult to define human's emotion, because we humans feel various emotions everyday and we have different feeling even if we meet the same situation. As time passes, a strong impression that we had at first will fade away. Unfortunately, our current techniques are not applicable to such changeable emotions. We should develop a new method to represent a growing or fading emotion in our mind. Although many problems for deep mental state still remain unresolved, our expectations are positive towards the development of some improved techniques in the near future.

References

Akamatsu, A., Sasaki, T., Fukamachi, H., and Suenaga, Y. (1993), "Automatic extraction of target images for face identification using the subspace classification method," *IEICE Trans.*, vol. E76-D, no. 10, pp. 1190-1198.

Ekman, P. and Friesen, W.V. (1975), *Unmasking the Face: a Guide to Recognizing Emotions from Facial Clues*, N.J.: Prentice-Hall.

Elliott, C.D. (1992), *The Affective Reasoner: a Process Model of Emotions in a Multi-agent System*, Doctor Dissertation, Northwestern University.

Fukumura, N., Uno, Y., and Suzuki, R. (1998), "Learning of many-to-many relation between different kinds of sensory information using a neural network model for recognizing grasped objects," *The Brain & Neural Networks*, vol. 5, no. 2, pp. 65-71. (In Japanese.)

Hevner, K. (1936), "Experimental studies of the elements of expression in music," *American Journal of Psychology*, vol. 48, pp. 246-268.

Hevner, K. (1937), "The affective value of pitch and tempo in music," *American Journal of Psychology*, vol. 49, pp. 621-630.

Ichimura, T., Ishida, H., Mera, K., Oeda, S., Yamashita, T., and Sugihara, A. (2001a), "An emotional interface with facial expression by sand glass type neural network and emotion generating calculation method," *Journal of Japanese Human Interface Societies*, vol. 3, no. 4. (In Japanese.)

Ichimura, T., Ishida, H., Terauchi, M., Takahama, T., and Isomichi, Y. (2001b), "Extraction of emotion from facial expression by parallel sand glass type neural networks," *Proc. of the 5th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES2001)*, vol. 1, pp. 988-992.

Ichimura, T., Mera, K., Ishida, H., Oeda, S., Sugihara, A., and Yamashita, T. (2001c), "An emotional interface with facial expression by sand glass type neural network and emotion generating calculation method," *Proc. of the First International Symposium on Measurement, Analysis and Modeling of Human Functions*, pp. 275-280.

Irie, B. and Kawato, M. (1990), "Acquisition of internal representation by multi-layered perceptrons," *IEICE Trans.*, vol. J73-D-II, no. 8, pp. 1173-1178. (In Japanese.)

Kawata, T. (1976), *Affine Geometry – Projection Geometry*. (In Japanese; *Iwanami Kouza Kiso Sugaku Senkei Daisu 5*), Iwanami Syoten.)

Melucci, M. and Orio, M. (1999), "Musical information retrieval using melodic surface," *Proc. of 4th ACM Conference on Digital Libraries*, pp. 152-160.

Minoshita, S., Satoh, S., Morita, N., Nakamura, T., Matsuzaki, I., Kikuchi, T., and Oda, S. (1997), "Assessing recognition of affects in facial expression through the use of Nohmen," *Japanese Journal of Ergonomics*, vol. 33, pp. 79-86. (In Japanese.)

Minoshita, S., Yoshikawa, M., Morita, N., Yamashita, T., and Satoh, S. (1999), "The relation between the recognition of facial expression and the employment of schizophrenics – using the score of the Noh Mask Test," *Proceedings for Second International Congress on Cognitive Science*, pp. 367-372.

Okada, N. (1996), "Integrating vision, motion and language through mind," *Artificial Intelligence*, vol. 10, pp. 209-234.

Osaka, N. (1986), "Cross-cultural differences in the perception of facial expressions of ambiguous Noh faces," *Bulletin of the Psychonomic Society*, vol. 24, pp. 427-430.

Plutchik, R. (1989), *Emotion: Theory, Research & Experience*, vol. 4 (The measurement of emotions), San Diego: Academic.

Quinlan, J.R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers.

Rolland, P.Y., Raskinis, G., and Ganascia, J.G. (1999), "Musical content-based retrieval: an overview of the melodiscov approach and system," *Proc. of 7th ACM International Multimedia Conference*, pp. 81-84.

Russell, J.A. and Bullock, M. (1985), "Multidimensional scaling of emotional facial expressions: similarity from preschoolers to adults," *Journal of Personality and Social Psychology*, vol. 48, no. 5, pp. 1290-1298.

Schlosberg, H. (1952), "The description of facial expressions in terms of two dimensions," *Journal of Experimental Psychological Review*, vol. 61, pp. 81-88.

Shirahama, N. (2000), "An approach to transitional rules of emotional words based on the theory for subjective observation model," *Proceedings of IEEE International Conference on SMC (SMC2000)*.

Shirahama, N. (2001), "A proposal of artificial emotion processing system and its color expression," *Proc. of the 5th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES2001)*, vol. 2, pp. 978-982.

Shirahama, N. and Miyamoto, K. (2001), "An artificial emotion model processing system and its color expression," *2nd International Symposium on Advanced Intelligent System Conference Proceedings*, pp. 267-270.

Tanaka, H. (1987), "Fuzzy data analysis by possibilistic linear model," *Fuzzy Sets and Systems*, vol. 24, pp. 363-375.

Tanaka, H. and Ishibuchi, H. (1991), "Identification of possibilistic linear systems by quadratic membership functions of fuzzy parameters," *Fuzzy Sets and Systems*, vol. 41, pp. 145-160.

Tanaka, H. and Watada, J. (1988), "Possibilistic linear system and their application to the linear regression model," *Fuzzy Sets and Systems*, vol. 27, pp. 275-289.

Taniguchi, T. (1995), "Construction of an affective value scale of music and examination of relations between the scale and a multiple mood scale," *Japanese Journal of Psychology*, vol. 65, no. 6, pp. 463-470. (In Japanese.)

Tseng, Y.H. (1999), "Content-based retrieval for music collections," *Proc. of 22nd International ACM SIGIR Conference*, pp. 176-182.

Tsuji, S. (ed.) (1997), *Kansei no Kagaku (Science of Kansei)*, Science publishers. (In Japanese.)

Ueki, N., Morishima, S., Yamada, H., and Harashima, H. (1993), "Expression analysis/synthesis system based on emotional space constructed by multi-layered neural network," *IEICE Trans.*, vol. J77-D-II, no.3, pp. 573-582. (In Japanese.)

Waley, A. (1997), "Hagoromo," The University of Virginia Library Electronic Text Center. Available at http://etext.lib.virginia.edu /japanese/noh/WalHago.html .

Xiao, Y., Chandrasiri, N.P., Tadokoro, Y., and Oda, M. (1998), "Recognition of facial expressions using 2-D DCT and neural network," *IEICE Trans.*, vol. J81-A, no. 7, pp. 1077-1086. (In Japanese.)

Yamashita, T., Minoshita, S., Morita, N., and Satoh, S. (1999), "Recognition of affects in the facial expressions of Noh mask by fuzzy reasoning," *Japanese Journal of Ergonomics*, vol. 35, pp. 193-199. (In Japanese.)

Yoshino, T., Takagi, H., Kiyoki, Y., and Kitagawa, T. (1998), "An automatic metadata creation method for music data and its application to semantic associative search," *Information Processing Society of Japan SIG Note*, 98-DBS-116(2), pp. 109-116. (In Japanese.)

Chapter 7

Public Opinion Channel: a Network-Based Interactive Broadcasting System for Supporting a Knowledge-Creating Community

T. Fukuhara, N. Fujihara, S. Azechi, H. Kubota, and T. Nishida

In this chapter we propose a system for supporting knowledge creation in a network community. Called a *Public Opinion Channel (POC)*, it automatically creates and broadcasts radio and TV programs based on messages received from community members, who can easily find differences in their viewpoints and opinions because the programs broadcasted by a POC include minority opinions as well as majority ones. We discuss POC design concepts and a prototype system from the viewpoints of social psychology and cognitive psychology. From the viewpoint of social psychology we pointed out obstacles to smooth discussions in a community and propose the concept of a *dry community*, where only the logical contents of messages are exchanged. From the viewpoint of cognitive psychology we explain why people have difficulties creating knowledge and we propose to use *metacognition*, which enables them to find differences in their thoughts and viewpoints. The requirements derived from these psychological viewpoints have been implemented in a prototype POC system, and the evaluation of that system has provided guidelines to the design of communication tools for knowledge-creating communities.

1 Introduction

The advent of the Internet has removed temporal and spatial constraints on communication and brought about *network communities* formed and maintained on the global information network. The *community knowledge* constructed by community members spontaneously and cooperatively is an important asset of any network, and it comprises not only shared documents but also free softwares, collective decisions, consensus, and shared beliefs and viewpoints.

A network community is called a *knowledge-creating community* if it explicitly or implicitly emphasizes the creation and maintenance of community knowledge. Typical examples of knowledge-creating communities sharing and exchanging specialized knowledge among knowledge workers can be found in education and business. People in other domains, such as nonprofit organizations (NPOs) or local communities, are learning from each other. They are also members of knowledge-creating communities.

Members of successful knowledge-creating communities exchange information and knowledge effectively and thus create community knowledge synergetically. However, knowledge-creating communities are not always successful. Knowledge creation in unsuccessful communities is hindered by frequent *flaming* that discourages community members' impulses toward creative discussion.

The goal of our research is to understand and develop a communication tool that a knowledge-creating community can make community knowledge more effectively. We would like to find out what aspects of interactions facilitates the knowledge-creating process in a community and to apply the insights to the development of an effective communication medium.

We argue from a social psychological viewpoint that people tend to be confused by incompletely communicated personal attributes such as social position, age, and gender, we introduce the *informational humidity model* to characterize the influence of message structure on the nature of social interaction in a community. We point out that a knowledge-creating community can maintain creative discussion only when the participants in a discussion exchange logical information rather than the personal attributes of the people providing that information.

From a cognitive psychological viewpoint we emphasize the importance of *metacognition*, which frees people from the bounds imposed by their own particular thinking styles. We argue that metacognition increases the number of opportunities for community-wide conversations that make community members aware of their differences of opinions and different beliefs.

The central contribution of this chapter is the Public Opinion Channel (POC), a novel interactive broadcasting system designed to help a community formulate public opinions. A POC continuously collects messages from members in a community, automatically creates multiple threads of stories together with headlines and summaries reflecting the content of those messages, and broadcasts those stories within the community. It not only supports community-wide discussions but also serves as a facility accumulating small talk at the daily-life level so that community members can build a common ground. Messages from community members are incorporated in a thread of stories on the fly according to an automated publishing and editing policy. A POC receiver allows the user to receive messages in both passive and active modes. In the passive mode, POC broadcasting is just like an ordinary radio broadcasting, allowing the user to casually receive information at the background level of cognition. In the active mode, the user can focus on a subject of interest and retrieve information on demand.

The proposed framework is general enough to be applicable to a vast range of applications related to knowledge management as well as to those particularly relevant to learning communities and consumer communities.

A POC cannot be useful in knowledge sharing, discussion facilitation, and public opinion formation unless it protects the user from information overload, provides a fair ground for discussion, and encourages participation.

We are implementing an information summarization mechanism that can reduce information overload and give the user a fair overview of the accumulated information. We are also implementing a mechanism of incrementally organizing accumulated messages into multiple threads of stories so that the user can choose and follow an interested stream of information. This mechanism gives the POC a dynamic flavor, entertaining users whose information demands are always changing.

We have developed a prototype POC system, and here we give some implementation details and preliminary analysis results based on small-scale experiments. An interview-based method and social network analysis were used in the evaluation of those results.

This chapter is organized as follows. Section 2 describes the conceptual framework and functions of a POC. Section 3 analyzes issues related to network communication tools from the viewpoint of social psychology and proposes a communication model for facilitating knowledge creation in a community. Section 4 describes knowledge creation from the viewpoint of cognitive psychology and proposes a notion of metacognition that enables humans to change their ways of thinking. Section 5 describes our prototype POC system and Section 6 presents the results of preliminary evaluation experiments with that system. Section 7 discusses the informational humidity model from the viewpoint of social psychology and also discusses future work.

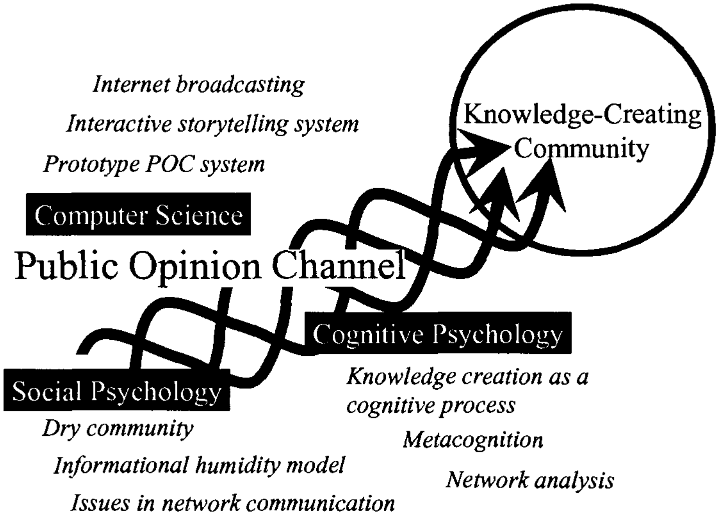

Internet broadcasting
Interactive storytelling system
Prototype POC system
Computer Science
Public Opinion Channel
Knowledge-Creating Community
Cognitive Psychology
Knowledge creation as a cognitive process
Metacognition
Network analysis
Social Psychology
Dry community
Informational humidity model
Issues in network communication

Figure 1. Overview of this chapter. This figure shows the collaboration between different research areas. Research themes related to a POC are tightly coupled and that aspects of a POC may attract the attention of readers interested in social psychology, cognitive psychology, or computer science.

The structure of issues raised in this chapter is shown in Figure 1. Collaboration between different research areas such as social psychology, cognitive psychology, and computer science is necessary if we are to understand and support a knowledge-creating community.

2 Public Opinion Channel

A POC is an automatic interactive broadcasting system for supporting knowledge creation in a community (Azechi *et al.* 2000, Nishida *et al.* 1999, Nishida 2000). As shown in Figure 2, it comprises the community members as well as the POC system. The system input is messages gathered from members, and the output is a story summarizing messages. Because there is an enormous number of messages

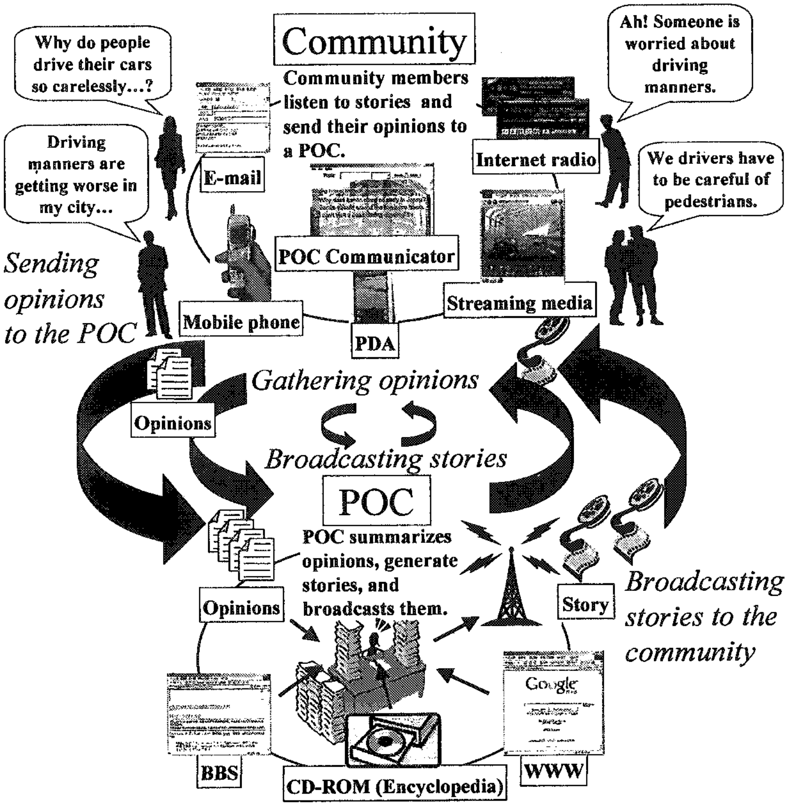

Community
Community members listen to stories and send their opinions to a POC.
Why do people drive their cars so carelessly...?
Driving manners are getting worse in my city...
Ah! Someone is worried about driving manners.
We drivers have to be careful of pedestrians.
E-mail
Internet radio
POC Communicator
Streaming media
Mobile phone
PDA
Sending opinions to the POC
Gathering opinions
Broadcasting stories
Opinions
POC
POC summarizes opinions, generate stories, and broadcasts them.
Story
Broadcasting stories to the community
Opinions
BBS
CD-ROM (Encyclopedia)
WWW
Google

Figure 2. Conceptual framework of a POC. A POC consists of community members and a POC system that makes a story based on messages gathered from members and broadcasts it on the Internet as a TV and/or radio program. Members can easily listen to or watch the story by using either a PC or a network-connected TV or radio, a mobile phone, or a personal desktop assistant (PDA).

in a community, it is difficult for members to know who thinks what. The POC system therefore gathers and summarizes them, classifying them into categories and integrating the important messages in each category into a comprehensive story. Some stories include related information from an encyclopedia on a CD-ROM or a DVD-ROM, Web pages, a mailing list, and a discussion log such as the archives of a bulletin board system (BBS).

One of the major differences between a POC and media already available, such as TV and the radio, is that the POC gathers the trifling thoughts and opinions of community members. Someone who has an opinion on driving manners, for example, has no effective ways to express that opinion on the available media. TV and radio stations rarely broadcast such opinions because there is little interest in them. Even if people with opinions write them on their Web pages, those page would be buried under other Web pages in the retrieval results of Web search engines. A POC, however, accepts such opinions and broadcasts stories featuring them. Because it considers a body of opinions, it summarizes the majority opinions in a community and also circulates minority opinions. Thus, the trifling thoughts and opinions of community members are circulated much more widely than they are on the media already available.

A POC system and its community members interact in the following four stages:

1. Call-for-opinion stage
2. Broadcasting stage I
3. Feedback stage
4. Broadcasting stage II

2.1.1 Call-for-Opinion Stage

In this stage the POC system announces a theme for upcoming stories, and community members send theme-related messages to the system. Members can also propose another theme to the system. Because members can use various communication tools such as mobile phones, PDAs, and PCs, they can send their opinions from anywhere. Students can send messages from a classroom by using network-connected PCs, and people in the street can send their messages from a cafe or a tram by using their mobile phones.

2.1.2 Broadcasting Stage I

In this stage the POC system makes a story that features the theme and broadcasts the story to the community. After the system first excludes junk messages that include malicious words, it gathers theme-related information from the Web and encyclopedias and then system integrates the messages and the related information into a story.

The stories are broadcasted on various media such as streaming video and audio on the Internet and are introduced by virtual newscasters and disc jockeys on TV and radio programs. Some stories are interactive ones in which community members can ask the virtual newscasters for details and related information and thereby change the course of the stories.

2.1.3 Feedback Stage

In this stage community members send the POC system their comments on the stories and their scores for the stories. Some might send supplemental information, and others might suggest related themes. This information and these suggestions are utilized for updating the stories. Some comments are incorporated into upcoming stories and others are stocked for making a story.

The POC system also takes account of the scores that members send, and stories that obtained high scores are broadcasted repeatedly and stories that obtained low scores are removed from the list of programs.

2.1.4 Broadcasting Stage II

In this stage the POC system updates an existing story on the basis of comments. After supplemental information is added to the story and a new viewpoint is introduced, the system broadcasts the updated story to the members. The members listen to and watch the story again, and they again send the POC system their responses to the story.

Through these stages the POC system and community members evolve stories incorporating the personal knowledge of each member, thereby forming community knowledge.

3 POC Characteristics from the Viewpoint of Social Psychology

The informational humidity model (Azechi 2000a) divides information into two classes: *dry information* and *wet information*. Dry information is content-oriented information, and wet information is personality-oriented information. The troubles in existing network communities, such as flaming, is caused by the wet information. Dry information, on the other hand, is suitable for smooth communication between members and for knowledge creation.

A POC is designed to create a *dry community* where only dry information is exchanged. A dry community supports knowledge creation by facilitating free discussion. Although an anonymous community on the Internet often causes problems due to the irresponsibility often associated with anonymity, a POC can prevent those problems.

3.1 Problems with Communication Tools

A general problem for those who design and develop new communication tools on the Internet is that users often lose their desire to use the tools (Azechi and Matsumura 2001) and are reluctant to express their opinions.

The reason users become discouraged are due to the following three problems:

1. The cognitive burdens due to expressing information by using communication tools.
2. Anxiety about being evaluated by others.
3. The abusing or flaming effect.

The first problem is related to human cognitive ability: expressing opinions and information clearly is too hard for most people. Even formulating original, novel, interesting, and consistent opinions requires a lot of cognitive work. People thus get tired immediately and give up expressing their opinions.

The second problem is that people who express their opinions are always anxious about being evaluated. People are often afraid of the judgments of others regardless of whether their opinions are accepted or rejected. This is because an acceptance and a rejection directly affect their identity and self-esteem. This is a fundamental emotional response of people in a society, especially when they communicate with each other.

The third problem is that Internet communication tools tend to exacerbate verbal abuse, or the *flaming effect* (Lea *et al.* 1994). The flaming effect is characterized as an unlimited 'slander battle' and is rarely seen in face-to-face communication. People who use communication tools are often afraid of being abused by other people.

One of causes of the flaming effect is anonymity. On the Internet, people often hide their real names and personal information by using fictitious names. Furthermore, some people behave as if they are different people on the Internet for hiding their personalities in the real world. This kind of anonymity on computer-mediated communication (CMC) often triggers inappropriate behaviors that in the real world are kept under control.

The questions we need to answer can be summarized as follows:

- How do we solve these three problems?
- How do we encourage community members to use communication tools?

A dry community could solve these problems, and the following subsection describes the informational humidity model that makes a community a dry community by using a POC.

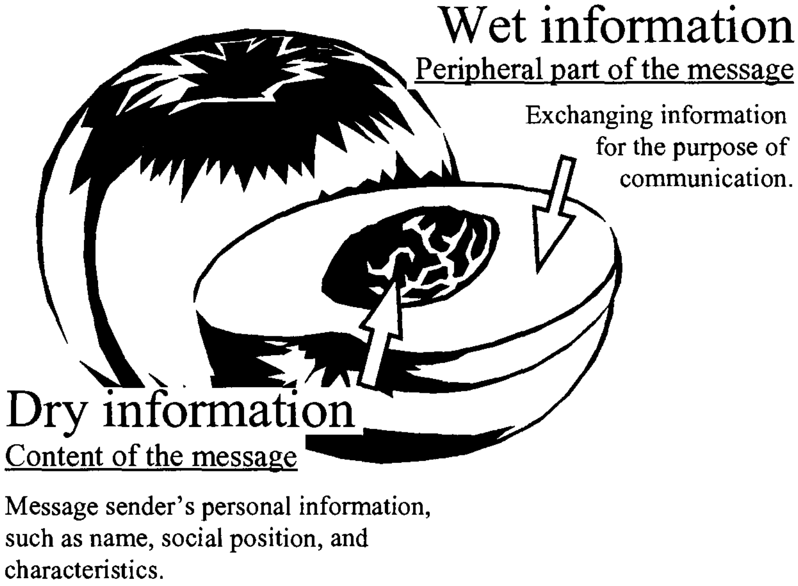

Wet information
Peripheral part of the message
Exchanging information for the purpose of communication.
Dry information
Content of the message
Message sender's personal information, such as name, social position, and characteristics.

Figure 3. Image of the informational humidity model. A message has two-class structure, and the two classes are like the two parts of a fruit: wet information corresponding to the edible part, and dry information corresponding to the seed part.

3.2 Informational Humidity Model

The informational humidity model is an inclusive communication-community model. In it, the messages people send are classified as dry information and wet information. This model suggests that communication tools encouraging free discussion should be designed to make the community a dry one. A dry community can sustain subject-concentrated discussions consisting of dry information.

This model classifies the parts of a messages into two classes of information: dry and wet (Figure 3). The dry information is the content of the message itself. It is usually linguistic information and is processed logically. It contains a description of a fact, an evaluation for the fact, and information about agreement or disagreement with other massages. Dry information is closely related to the subject of a discussion.

Wet information, on the other hand, is the peripheral part of message that includes personal information about the message sender — the name, social position, and characteristics of the sender; and, sometimes, "emoticons" (punctuation marks arranged in patterns to show emotional states). The major function of wet information is to smooth conversation and indicate intention. Wet information works well in face-to-face communication because there one can transmit one's wet information effectively. But when CMC tools such as an e-mail system and BBS are used, they often trigger verbal abuse and the flaming effect because they cannot transmit wet information effectively.

Wet and dry information are closely connected to two functions of a group process. Dry information corresponds to a *performance function* of a group that regards the performance of a task as important (Cartwright and Zander 1968). For example, members of a group in which they concentrate on solving intellectual problems or facing difficult tasks tend to exchange dry information. Wet information corresponds to a *maintenance function* of a group that regards the maintenance of a relationship between group members as important (Cartwright and Zander 1968). For example, members of a group in which they want to become friends or confront an enemy tend to exchange wet information.

A dry community, which is characterized by exchanging dry information, encourages its members to express their opinions on the Internet. The interests of the members in a dry community are mainly directed to the performance of a task, and a discussion in a dry community concentrates on the subject of the discussion. Members who join the discussion are interested in an argument of an opinion and are not interested in the name of the author of an opinion. This is because only an argument contributes to solving a task. A dry community is highly a democratic community, and its communication tools should hide all wet information indicating the message sender's name and social position.

A dry community also solves the three problems of network communities described in Section 3.1. The first problem is solved because the people in a dry community can easily express their preliminary opinions to the community. The second problem, anxiety about evaluation, is also solved because people who express their opinions are never evaluated by others. This is because no one can know who expressed those opinions. The people in a dry community can therefore express their opinions freely. The third problem, a flaming effect, is also unlikely because no one in the dry community can identify the author of an opinion.

Is the wet information not essential to a network community? No. It does, however, contribute to the maintenance function of a group. And if the communication tools a group used could control the wild nature of the dangerous wet information, it would be useful for connecting human relationships in that group. This kind of community, called a *wet community*, would contribute to creation of knowledge in a way different from that in which a dry community does.

The characteristics of dry and wet communities are listed in Table 1. A dry community is appropriate for a free discussion among a rather large number of members who are highly mobile and have clear intentions. Wet information is filtered out of dry communities because it often discourages members from expressing their opinions. A wet community, on the other hand, is appropriate for creating the basis of discussion and establishing close relationships between community members. Although a precondition of any discussion in a wet community is that its members regard each other as reliable people with whom novel information can be exchanged, there is some danger that

Table 1. Characteristics of dry and wet communities.

	Dry community	Wet community
Purpose	Making a decision, Exchanging opinions	Establishing a friendship and partnership
Feature	Performance-oriented, temporary	Maintenance-oriented, permanent
Group size (number of people)	Small to large (10 – 1,000)	Small (10 – 100)
Knowledge	Explicit, descriptive	Tacit, procedural
Example	Public discussion of a policy or social event	Club, sport team, friends, family

the wet information harms the relationships of members. This means that some control of wet information is needed.

4 POC Characteristics from the Viewpoint of Cognitive Psychology

Does a POC really facilitate knowledge creation? And if so, what kinds of characteristics of a POC are responsible for this facilitation? This section examines these questions from the viewpoint of cognitive psychology.

4.1 Does a POC Facilitate Knowledge Creation in a Community?

To examine whether a POC facilitates knowledge creation, we need to evaluate the efficacy of a POC for knowledge creation in a community. Previous investigator, however, did not use methodologies that could evaluate the effects of communication tools working on network communities. We therefore had to develop suitable methods for evaluating the efficacy of communication tools.

4.2 Cognitive Psychological Research on Knowledge Creation

Knowledge creation is the process of generating new ideas and viewpoints based on preexisting knowledge and information, and many topics of cognitive psychological researches are related to a cognitive process of knowledge creation. Two of the most closely related topics are creativity and creative thinking. Guilford classified thinking into two subtypes (Guilford 1961): *Convergent thinking* is the process by which people make only one answer to a question, and *divergent thinking* is the process by which they make two or more answers to a question. Knowledge creation is not the simple application of past experiences to a problem situation, but is the generation of new solutions. *Creative thinking* is similar to divergent thinking. Various methods facilitating creative thinking have been proposed (Kawakita 1967), and one of the most well known is the *brainstorming* proposed by Osborn (Osborn 1953). It has some rules that facilitate thinking in a group: "Do not judge or criticize the ideas of other people." "Respect ideas that are free and bold." "Generate as many ideas as possible." "Combine and improve the ideas of other people." These rules can be regarded as basic to creative thinking, and we can use them to help us implement tools like a POC.

Metacognition is a topic relating to the creation of knowledge. It is defined as knowledge and cognition about cognitive objects and processes, and it can be divided into *metacognitive knowledge* and *metacognitive experience* (Flavell 1987) (Sannomiya 1995). Metacognitive knowledge includes information about the trend of a person's own thinking and the trend of other people's thinking and also includes information about the efficacy of strategies that people have already used to solve problems. Metacognitive experiences are conscious experiences about ongoing cognitive activities (i.e., monitoring thought processes, controlling activities, and adapting thoughts). To generate new ideas, we need to be free from the trends of our own ways of thinking and able to reconsider problems from various points of view. Metacognition enables us to change viewpoints to monitor cognitive activities and to control ways of thinking.

Psychological research has shown that it is difficult for us to change our points of view voluntarily, and Gick and Holyoak (Gick and Holyoak 1980) showed that it is hard for people to transfer previously learned strategic knowledge to solve a similar problem (Duncker 1945). The knowledge already acquired could be transferred only when people were explicitly informed that the two problems were similar and the previous knowledge could be used. Experiments on human cognitive categorization also have shown that people do not always use their knowledge voluntarily (Fujihara 1998) (Fujihara 2000) (Medin *et al.* 1987). The subjects in a psychological experiment reported by Fujihara (Fujihara 1998), for example, were shown examples from two categories and asked to learn both categories. They were then asked to categorize items that could be classified into the categories. Some of the items were the ones which participants had already learned and others were ones that contained new information and that participants could categorize if they made inferences about the two categories. The results of this experiment showed that the subjects generally did not make inferences voluntarily. For example, they used new information to categorize items only when they had previously been given information on category labels. Fujihara's work thus suggested that people seldom change their points of view voluntarily and that instructions from others can facilitate metacognitive experience.

Conversations with others and CMC can show us other people's ideas and viewpoints and give us the opportunity to change our own viewpoints. As mentioned above, people seldom change their points of view voluntarily. If communication tools like a POC support metacognitive activities, they could help us take into account other viewpoints and ways of thinking. What do such tools have to do? Fujihara suggested that one important function of a tool supporting metacognition is to show the differences between opinions of people explicitly (Fujihara 1999).

5 Prototype POC System

We have implemented a prototype POC system consisting of a *POC Server*, a *POC Communicator*, and a *POC caster* and *POC radio*. The POC server is a broadcast system for making and broadcasting stories based on gathered opinions, and the POC Communicator is a tool for browsing and editing opinions. Community members can send their opinions to the POC server, which gathers opinions and broadcasts them as a story. The *POC caster* and the *POC radio* are tools for watching the stories and listening to them as if they were TV and radio programs.

5.1 Overview

The prototype system takes a server/client model, i.e., it consists of a *POC server* and several *POC client tools*. The server gathers opinions from community members, generates stories, and broadcasts them within the community. The POC client tools are tools for browsing and sending opinions and for sending the signals that make it possible for community members to watch the stories and listen to them. The POC Communicator is a client tool for editing and sending opinions. The POC caster and the POC radio are client tools for watching the stories and listening to them. Members can create a new opinion and a story and post them to the POC server by using the POC Communicator and can watch stories and listen to them by using the POC caster and the POC radio.

5.2 POC Server

The main functions of the POC server is to make stories and broadcast them to the POC client tools. It automatically generates stories based on messages from community members and then broadcasts the stories to members.

5.2.1 Making a Story

The following is an overview of the story-making process.

1. Pick up a message (*source message*) from the message database.
2. Retrieve messages from the message database by using the title of the source message.
3. Sort retrieval results chronologically and add the first n messages to the source message.

A story consists of a source message and the n messages most relevant to the source opinion. The server first picks up a message from the message database where messages from community members are stored. This message is the introduction to the story. The server then retrieves related messages based on keywords in the title of the source message. We chose nouns in the title by using a morphological analysis tool. Finally, retrieval results are sorted chronologically and the most relevant n messages are added to the source opinion. An example of a story is shown in Table 2, where a presenter who is a newscaster or a disc jockey introduces two messages related to the

Table 2. Example of a story. It consists of messages retrieved from the message database and sorted chronologically to follow the flow of an original discussion.

Presenter	*I will show you messages accepted from you.* *The topic is "affordance".*
Message 1	Does anyone know about affordance? Do you have any recommendation for reading?
Presenter	*We have a related message.*
Message 2	Hi folks, I found a good textbook on cognitive psychology. The textbook describes affordance in detail.
Presenter	*Thank you all. We are waiting for your messages on "affordance."*

keyword “affordance.” These messages are sorted chronologically in order to follow the stream of a discussion.

A story has a *context* that is an order of sentences according to a sequences of topics, causes and results, and so on. We are implementing various context-generation methods into the process of story generation, one of which is a topic-based summarization method (Fukuhara 2000). This method generates a context consisting of related sentences. Sentences are linked on the basis of a *theme* and a *focus*. A theme indicates a subject of a sentence and a focus indicates a topic that is emphasized in the sentence. We identify a theme and focuses by utilizing the case grammar. In the case grammar, each clause in a sentence has a case that indicates its function relative to a verb. We specify a theme and focuses of a sentence by identifying a case for each clause.[1]

A context is generated by linking sentences by finding a pair of a focus and a theme between two sentences. We link two sentences when a focus of a sentence equals to a theme of another sentence. The following is the algorithm for context-generation. The input is a set of sentences, and the output is a context consisting of a set of relevant sentences.

1. Pick up a sentence and substitute this sentence for the *current_sentence*.
2. Substitute the *current_sentence* for the *context*.
3. Do until n sentences are found.
   (a) Pick up a focus of the *current_sentence*.
   (b) Pick up a sentence whose theme is the focus of the *current_sentence* and substitute the sentence for *next_sentence*.
   (c) Concatenate the *next_sentence* to the *context*, and substitute the concatenated sentences for the *context*.
   (d) Substitute the *next_sentence* for the *current_sentence*.

[1] We use Kurohashi and Nagao’s (1994) Parser to determine the cases of clauses in Japanese sentences. See http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/index-e.html.

Italicized terms indicate variables in the algorithm. An example of a context is shown in Table 3. The context is generated by linking a focus and a theme. We pick up a sentence and find a focus of that sentence. In this case, we pick up "wind power" as a focus and select a next sentence whose theme is that focus. We then pick up a focus ("alternative sources") from that sentence. A context is generated by repeating these procedures until n sentences are found (n is a specified threshold).

Table 3. Example of a context. Italicized terms are themes, and terms in curly brackets are focuses. A context is generated by linking a theme and a focus.

No.	Sentence	Theme	Focus
S1	The government expects {wind power} to be a feasible energy source	government	wind power
S2	*Wind power* is one of the most widely used {alternative sources}.	wind power	alternative sources
S3	One of the *alternative sources* is {geothermal energy}.	alternative sources	geothermal energy

5.2.2 Broadcasting Stories

The POC server broadcasts stories in audio streaming, and the stories are played by the POC client tools. The POC radio, which is one of the client tools, plays a story as a radio program by receiving an audio streaming file. The server generates an audio file (MP3 file) by using a text-to-speech (TTS) system and broadcasts the file by using an MP3 streaming server such as icecast.[2] In the file a virtual disc jockey introduces members' messages according to the story. The disc jockey also plays music between stories. Community members can listen to the program by using MP3 players such as WinAmp[3] and XMMS.[4]

[2] http://www.icecast.org/

[3] http://www.winamp.com/

[4] http://www.xmms.org/

5.3 POC Client Tools

We have developed three types of POC client tools — a POC Communicator, a POC caster, and a POC radio — so that the user can access a POC in various situations.

5.3.1 POC Communicator

The POC Communicator comprises a message browser, message editor, and story editor.

Message browser

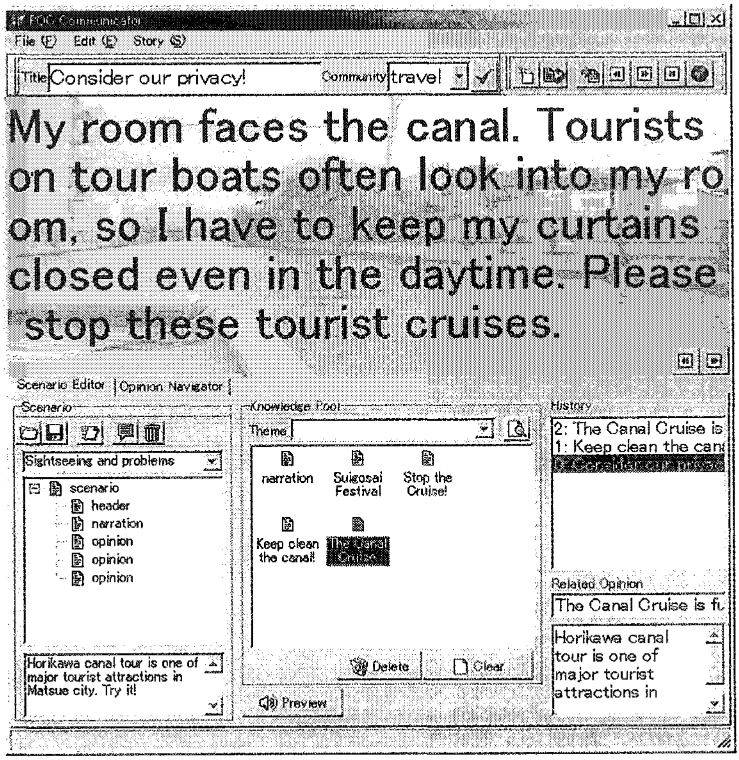
The message browser displays messages automatically. Community members can retrieve messages and browse retrieval results. Messages are circulated in the browser; that is, they are displayed repeatedly. Members can select the order in which they are displayed by choosing "random", "ascending" (chronological), or "descending" (the latest message comes first). Members can capture messages displayed on the browser and store them in the *personal knowledge pool* where messages are stored in local hard disks.

Message editor

The message editor enables members to edit and send a message to the POC server. A message consists of a title, a body, and a related URL. Members can also refer to other messages.

Story editor

The story editor enables members to edit a story by using messages. Members can create a story by making a *story tree* representing the structure of a story, and they can edit a message and add the message as a node of the tree. Members can also edit captured messages and add them to the tree. The story is sent to the POC server and broadcasted by the server.



Title: Consider our privacy!

Community: travel

My room faces the canal. Tourists on tour boats often look into my room, so I have to keep my curtains closed even in the daytime. Please stop these tourist cruises.

Sightseeing and problems

- scenario
  - header
  - narration
  - opinion
  - opinion
  - opinion

Horikawa canal tour is one of major tourist attractions in Matsue city. Try it!

narration | Suigosai Festival | Stop the Cruise! | Keep clean the canal! | The Canal Cruise

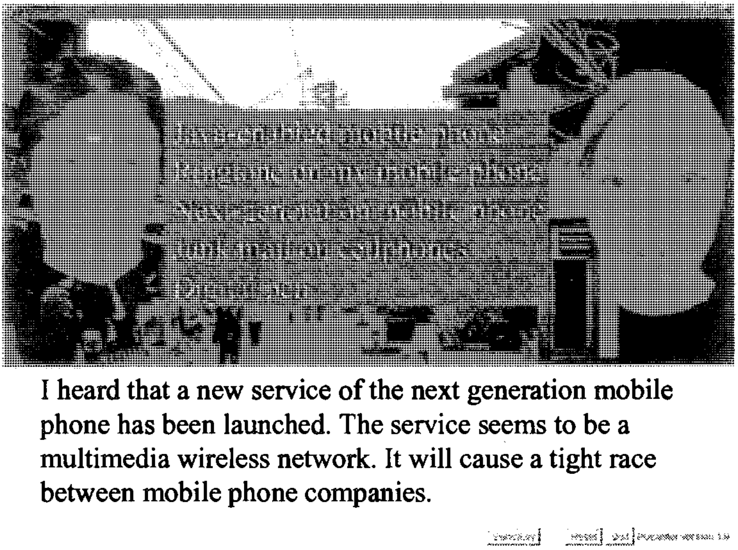
2: The Canal Cruise is
1: Keep clean the can

The Canal Cruise is fu

Horikawa canal tour is one of major tourist attractions in

Figure 4. Screen image of the POC Communicator. Community members can browse and edit messages. The upper part of the window shows a message browser. The lower part consists of story editor (left), personal knowledge pool for storing captured messages (center), and browsing history of messages (right). Members can capture messages and store them in the knowledge pool. They can also create a story by editing the captured messages. The story is sent to the POC server and broadcasted by the server.

Figure 4 shows a screen image of the POC Communicator. A message is shown on the upper part of the window, and a story tree, captured messages, and a browsing history of messages are shown on the lower part. Messages are displayed by the browser automatically, so members can browse messages by just looking at the browser.



I heard that a new service of the next generation mobile phone has been launched. The service seems to be a multimedia wireless network. It will cause a tight race between mobile phone companies.

Figure 5. Screen image of the POC caster. Virtual newscasters on both sides tell stories based on the topics appearing in the center of the screen. Subtitles on the lower part of the screen show the texts of the story. The newscasters talk about each topic to each other.

5.3.2 POC Caster

The POC caster[5] is an interactive storytelling system that shows a talk show with a story. Figure 5 shows a screen image of the system. There are two virtual newscasters, one on each side of the screen. The virtual newscasters tell a story based on the topics that appear in the center of the screen.

The POC caster is based on the *EgoChat* system, which is a conversational storytelling system. An EgoChat user can ask conversational agents a question by simply saying the question aloud. The agents tell each other stories related to the question, and the user can listen to the stories (Kubota *et al.* 2000).

[5] A paper by Hidekazu Kubota has been submitted to the journal of Japanese Society for Artificial Intelligence.

One of the major features of the POC caster is that it makes a *conversational scenario*, which is a script consisting of several related messages. The script is converted into a conversational representation in which one newscaster talks and the other replies, and vice versa. An example of a conversational scenario is shown in Table 4, where the newscaster A asks B to explain the meaning of the previous sentence. Then newscaster B explains the meaning.

A conversational scenario is generated by converting sentences in messages to a conversational representation. The POC caster has a knowledge-base in which phrase patterns that indicates intentions of sentences are described. By using the knowledge-base, the POC caster analyzes the intention of sentences. When a sentence that matches the patterns is found, the POC caster inserts sentences requiring an explanation, a description, and related information.[6] By applying the patterns to sentences in messages, POC caster generates a conversational scenario.

Table 4. Example of a conversational scenario. A conversational scenario is generated by editing an original message.

Original message

The Akishino River is popular among walkers in Nara.
There are many famous temples along the Akishino River.
What would you say to a walk?

Conversational scenario

Newscaster A	The next topic is walkers in Nara.
Newscaster B	The Akishino river is popular among walkers in Nara.
Newscaster A	What does it mean?
Newscaster B	There are many temples along the Akishino river.

[6] Examples are the sentences "What does it mean?", "Tell me more about it", and "Are there any related stories?"

5.3.3 POC Radio

The POC radio is a tool for listening to stories. By using MP3 players, community members can listen to stories broadcasted as a radio program by the POC server. In the radio program, a virtual disc jockey reads messages according to a story by using a TTS and plays music stored in the server. Stories are made on the POC server and broadcasted by an MP3 streaming server.

6 Evaluation of the Prototype POC System

To find out how the prototype POC system supports knowledge creation in an actual community, we carried out the following two experiments.

- How does the prototype POC system work in an actual community?
- How to evaluate an effect of a POC on knowledge creation?

Results of the experiments and findings from the results are described.

6.1 How Does the Prototype POC System Work in an Actual Community?

We interviewed six people who had used the prototype POC system for more than four weeks. They were not naive subjects but were members of the POC research group comprising two computer scientists, two psychologists, and a secretary (Table 5). Each was interviewed for 15-20 minutes.

We prepared a five-question questionnaire to make clear a user's motivation. All the subjects were not asked all the questions; the questions were only used as cues to draw responses from the subjects. The

Table 5. Constituents of the POC research group.

ID	Age	Gender	Role in the group
S1	28	Male	Computer scientist
S2	31	Male	Cognitive psychologist
S3	31	Female	Secretary
S4	26	Male	Computer scientist
S5	31	Female	Social psychologist
S6	30	Male	Social psychologist

Table 6. Questions asked in interviews.

How frequently do you use the prototype system?
How many messages do you write a day?
How do you use the system? Are you motivated to use the system?
How do you feel about the anonymity of a POC community?
Do you think that you can share information with other POC users?

questions are listed in Table 6, and the results of these interviews are described in the Appendix.

Our findings can be summarized as follows.
1. Few subjects were motivated to use the prototype system.
2. Junk messages discouraged users.
3. The POC Communicator seems to transfer wet information.

The subjects did not seem eager to use the prototype system. S2, for example, said, "I don't bother to use the prototype POC system continuously because the information and opinions available on it are not of interest to me." All the subjects nonetheless continued to use the system for at least one month after the interview. Although they were members of the development team and thus were externally motivated to use the system by force by oneself, it seems they also had some tacit motivation to use the system continuously.

Junk messages discouraged the subjects from using the system. S4 said "I found some interesting messages on the system, but there are also many junk messages". Junk messages (messages with information and opinions not directly linked to the subject under discussion) made it difficult for users to follow the subject of a discussion. Consequently, users lost their interest in the discussion.

Finally, the POC Communicator seeds to transfer wet information. Interviewed subjects said that they inferred the sender's intention behind the message. Although this intention is not wet information in the original sense of the term, it should be considered a kind of wet information. This phenomenon (inference of a message sender's intention) should be further investigated.

6.2 How to Evaluate the Effect of a POC on Knowledge Creation?

To evaluate communication tools, we should set up an appropriate *control condition* as a baseline. If this control condition is biased, the effect of the tools cannot be evaluated accurately. Although some researchers have evaluated tools without setting a control condition, they were not able to evaluate whether the tools actually support intelligent activities.

How should a control condition be set up? One appropriate way is to design the communication tools as a combinations of a basic part and some additional parts. The case in which people use only the basic part can be the control condition, and the cases where people use the tools constructed with a basic part and additional parts can be the experimental conditions. The effects of the communication tools could be discerned by evaluating the difference between the results obtained in the control condition and the results obtained in the experimental conditions. In the case of the prototype POC system, the case where the system that has basic functions is used could be set up as a control condition. The experimental conditions would be ones in which the system had the basic functions and one or more additional functions. Some POC research issues are "How should information be summarized to facilitate knowledge creation in a community?" and "Can anonymous communication systems inhibit communication problems like flames?" One possible experimental conditions would provide a POC with a summarization function, the effect of which could be evaluated by comparing the differences of some measurements (e.g., the frequency of use or the number of messages in circulation) between the control condition and the experimental condition. Different modules implementing the same function could also be compared this way.

Communication tools are not always designed with modules, however. Another way to set up a control condition would be to use typical situations in which people use ordinary network media like mailing lists, BBS, and chats. For this purpose it would be useful to define typical situations and to standardize procedures to collect data and to analyze data. Fujihara and Miura, for example, analyzed the behavior of search engine users (Fujihara and Miura 2000). They proposed categories to describe information searching behaviors by using the WWW search engines. Such research would reveal our common activities in network communities and would give us a baseline for evaluating new network communication tools.

Evaluation methods are roughly classified into these three types:

1. Analysis of users' subjective evaluations collected through the use of questionnaires and interviews.
2. Systematic experimental methods.
3. Analysis of user behavior observed in ordinary and natural settings.

Each of these methods has strong points and weak points, so the effects of a tool should be evaluated by using two or three types of methods together. Although some researchers evaluate applications by analyzing only the subjective judgments of users, the result of such evaluations should be considered critically.

Network analysis is a method, mainly used in sociology, for analyzing relationships between community members and relationships between companies. It describes networks as graph structure in which each node represents a person or a company and each link represents the relation between people or companies. It is used to investigate the structures of networks, the effects of network structures on community members, and the mechanisms of those effects.

One of the representative quantification methods is "degree" (Figure 6). Degree means the numbers of links each node has. There are two types of degrees: (1) *in-degree* that is the number of incoming links from other nodes into a node, and (2) *out-degree* that is the number of outgoing links from a node. As the case of Figure 6, in-degree of the marked node is 3 (i.e., there are three incoming links from other nodes to the marked node) and its out-degree is 2 (i.e., there are two outgoing links from the marked node to other nodes).

Figure 7 represents part of the network structure based on messages on the POC Communicator. Each square represents a message (a node), and the numbers written in squares represent the ID numbers of messages. The smaller the ID number is, the earlier the corresponding message were sent to the POC server.

Twenty nodes had no links to other nodes, and some constructed very simple links described in the lower part of the Figure 7. And sixty-five nodes constructed a highly complex network described in the upper part of the figure. Some nodes such as ID number 81, 82, and 97 plays important roles in the network because those nodes have many incoming links. An incoming link to a node indicates that a message represented as the node is referred to from other messages. We quantified these networks by degrees. To evaluate whether the prototype POC system facilitates community knowledge, we have to make some assumptions. One assumption is that conversation and communications on the system are more effective when there are more nodes hat have a large number of links. Figure 8 compares numbers

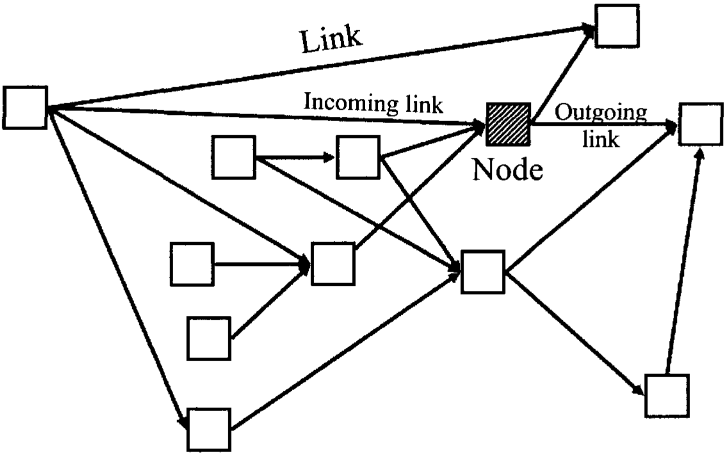


Figure 6. Example of a network. The network consists of nodes and directed arrows. A directed arrow from one node to another means that there is a link from the former node to the latter one. In-degree is the number of incoming links to a node and out-degree is the number of outgoing links from a node. For example, In-degree of the marked node in this figure is 3 and its out-degree is 2 because there are three incoming links to the marked node and two outgoing links.

between the in-degree and out-degree of each node in the network structure illustrated in Figure 7. In Figure 8 the average number of degrees was 8.8 and there were five nodes having a number of degrees more than 2.0 standard deviations greater than the average (in this case, $8.8 + 2.0 \times 8.9$). This looks slightly larger than the number expected to occur by chance (2.28 nodes).

This analysis is just a first step of our trial, so we have a lot of issues to discuss. First, we have to set an appropriate control condition. We are now using network analysis to analyze the network structure gained at a discussion on the POC Communicator. This network structure can probably by used as a baseline. Second, there are other possible ways to evaluate how well the prototype POC system facilitates knowledge creation. For example, messages could be clas-

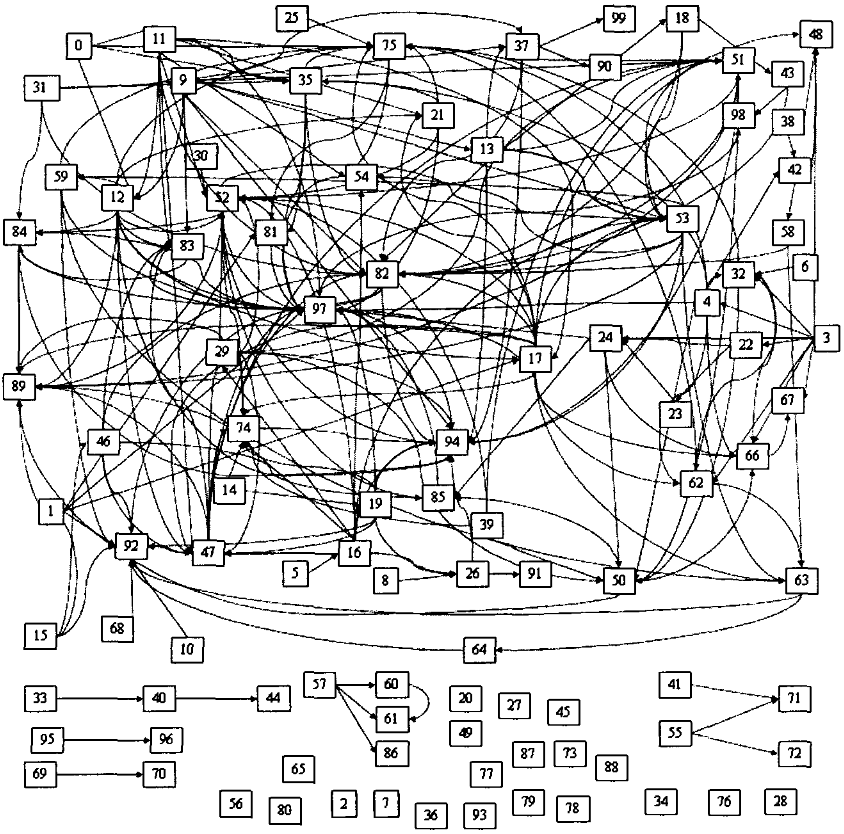


Figure 7. Network structure gained at a discussion on the POC Communicator. A square shows a message (a node) and an arrow shows a link between nodes.

sified into some clusters based on degrees. If there were links that connected messages from different clusters, those links may indicate that the system facilitates knowledge creation. The numbers of links connecting chronologically separated messages may also be an index of knowledge creation. We think the network analysis gives us an interesting viewpoint from which to evaluate communication tools on the Internet.

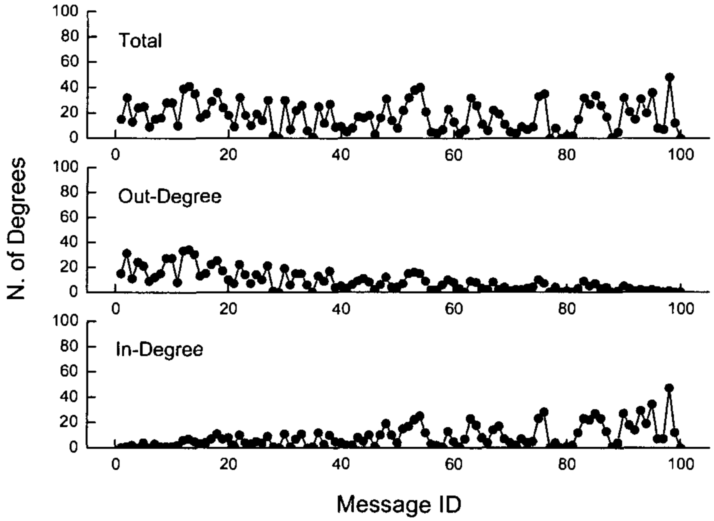


Figure 8. Comparing numbers between the in-degree and the out-degree of each node described in Figure 7. This figure shows how messages refer to/from others. We found that five messages were key messages for facilitating active discussion.

7 Discussion and Future Work

In this section, we discuss our proposals from the viewpoint of social psychology and also discuss future work.

7.1 What Kind of Communities Need a Dry Community?

Nonaka argued that both dry communication and wet community are needed for supporting knowledge creation in a community (Nonaka and Takeuchi 1995). He argued that both are useful for the process of knowledge creation.

We think that a dry communication is needed more in a large community than a small one. There are many communities that should be dry, such as a city community discussing city issues among citizens, a country-wide community discussing national policy, and an international community discussing environmental issues, the food problem, international disputes, and so on. In these large and open communities, wet information is useless in most cases because providing one's name, occupation, and nationality is of no significance to most community members. Those kinds of wet information, in fact, damage the relationships of members in a community.

7.2 Future Work of a POC from the Perspective of a Dry Community

To make a better dry community that encourages its members to express their opinions, the prototype POC system needs two additional functions: (1) a function for filtering out wet information, and (2) a message mediator system that expresses community members' opinions on behalf of themselves. Filtering out wet information will increase the users' tacit motivation. Filtering out junk messages would help prevent users being discouraged from expressing their opinions. Filtering out abusive and repeated information is also important in preventing problem due to the anonymity of a dry community.

A message mediator system moderates the bad behavior due to the inference of wet information (Figure 9). If the behavior of the members of a dry community is inevitably affected by inferred wet information, which is likely because this inference seems to be a fundamental human cognitive act, problems due to these bad behaviors can be avoided by providing a virtual target for the reactions elicited by wet information. If the mediator system provides opinions and broadcasts edited stories based on those opinions, members can infer only the mediator's wet information. This inferred wet information would not cause any trouble for the other members. The mediator system would thus help make a more effective dry community.

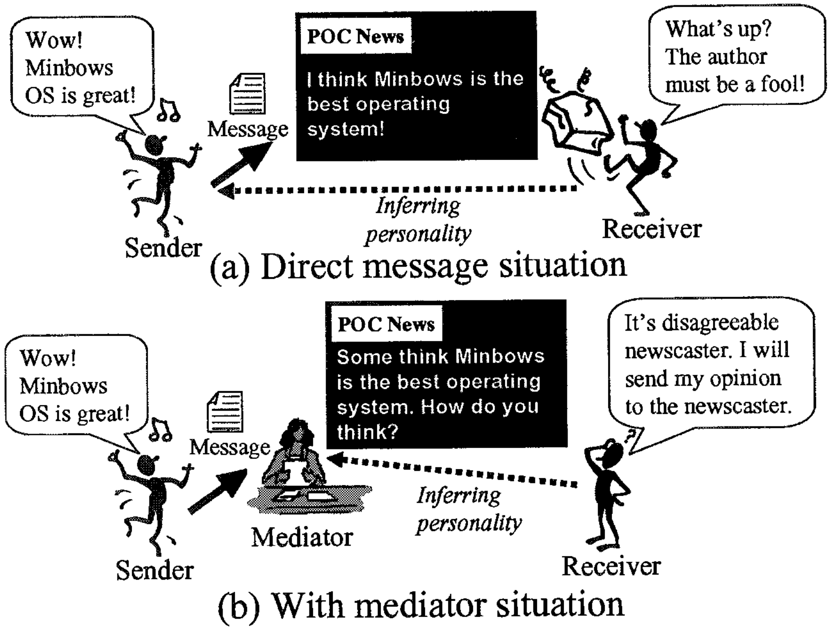

Wow!
Minbows
OS is great!
Message
POC News
I think Minbows is the best operating system!
What's up?
The author
must be a fool!
Inferring personality
Sender
Receiver
(a) Direct message situation
Wow!
Minbows
OS is great!
Message
POC News
Some think Minbows is the best operating system. How do you think?
It's disagreeable newscaster. I will send my opinion to the newscaster.
Inferring personality
Mediator
Sender
Receiver
(b) With mediator situation

Figure 9. Concept of the messages mediator system. The mediator system avoids the bad effects of wet information because community members who receive messages cannot directly infer the personal information of the actual message sender. They can infer only the (virtual) mediator's wet information.

7.3 Extension of the Informational Humidity Model

One extension of informational humidity model predicts that inferred wet information might cause problems not predicted by the original model (Azechi 2000b). Here we extend the framework of the informational humidity model by referring to Newcomb's social interaction model (Newcomb 1953) as shown in Figure 10. Newcomb's model contains three targets: (1) a message receiver A, (2) a message sender B and (3) a message subject X. This model merely indicates the relationships of A, B and X. For example, if the message receiver

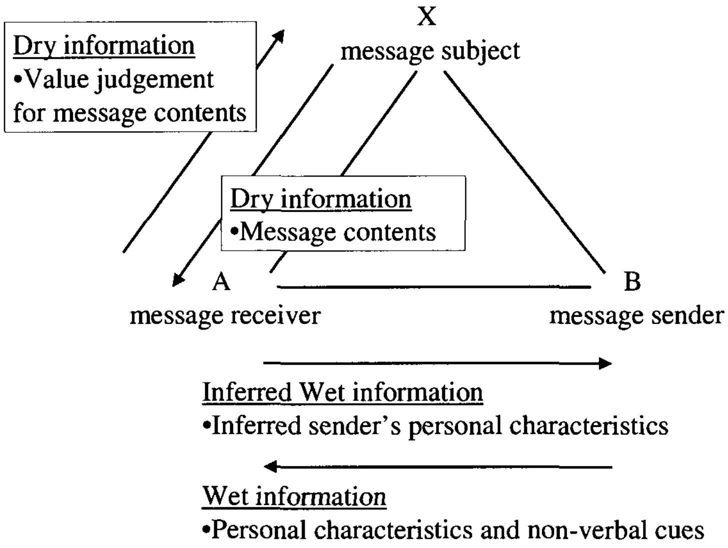


Figure 10. Extended informational humidity model: This figure indicates the relationships of the message receiver A, the message sender B, and the message subject X. A infers B's personal information even when it is not transmitted by B.

has negative impression to the message sender, s/he will interpret the subject negatively.

In the Figure 10, the arrows towards the message receiver (B to A and X to A) mean that information is being transferred. The information transferred from X to A, the message content, is the dry information; and the information transferred from B to A, the presentation of the sender's personal characteristics and non-verbal cues, is the wet information in the original sense. The arrows from the message receiver (A to B and A to X) respectively mean how and what the receiver infers and interprets with regard to the sender's character and with regard to the message content. The information whose transfer is indicated by those arrows is considered secondary wet information. The arrow from A to X represents the judgment of the value of the message, and the arrow from A to B represents the inferred characteristics of the message sender that is a new type of wet information, *the inferred wet information.*

We predict that the inferred wet information has another kind of anonymity effect. We worry about the flaming effect that might result from the inferred wet information. For example, if one infers that another person's personality is very bad because s/he expresses an opinion contrary to one's own opinion, one can embarrass him or her by posting malicious opinions. Thus, simply filtering out wet information and merely creating dry information would cause another type of problem for communication.

If we want to prevent problems caused by inferred wet information, we should know how it affects the behavior of community members. The results of a social psychological experiment that investigated how people in an anonymous community perceive other people would tell us something about how inferred wet information influence the behavior and motivation of people in a dry community. According to the findings of group dynamics, people tend to evaluate out-group members as less worthy than in-group members because they know less about the out-group members (Sharif *et al.* 1966). In the anonymous situation of the dry community, people are also assumed to think that fewer out-group members than in-group members express opinions contrary to theirs, so they think the contrary opinions are repeatedly expressed by a few people.

8 Conclusion

In this chapter we propose a POC for supporting knowledge creation in a community. There are many questions that need to be answered if we are to understand and support a knowledge-creating community, including what knowledge creation is, why communication tools have difficulties supporting it, and what features and functions are needed for the tools. To answer these questions, we propose a notion of a dry community from the viewpoint of social psychology. And from the viewpoint of cognitive psychology, we propose a notion of metacognition that enables us to change our ways of thinking and create knowledge. We have implemented a prototype POC system consistent with these proposals, and our preliminary experiments with that system have provided us with guidelines for designing a communication tool for supporting a knowledge creation in a community. We think we will be able to design better network communication tools by combining the aspects of knowledge-creating communities evident from different viewpoints.

References

Azechi, S. (2000a), "Social psychological approach to knowledge-creating community," in Nishida, T. (ed.), *Dynamic Knowledge Interaction*, pp. 15-57, CRC Press, Florida.

Azechi, S. (2000b), "Dry information, communication, and communities," in Baba, N., Jain, L.C., and Howlett, R.J. (eds.), *IEEE Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES2000)*, vol. 1, pp. 72-75, IEEE Press, USA.

Azechi, S., Fujihara, N., Sumi, K., Hirata, T., Yano, H., and Nishida, T. (2000), "Public Opinion Channel: a challenge for interactive community broadcasting," in Ishida, T. and Isbister, K. (eds.), *Digital Cities: Technologies, Experiences, and Future Perspectives*, pp. 427-441, Lecture Notes in Computer Science (vol. 1765), Springer-Verlag.

Azechi, S. and Matsumura, K. (2001), "Motivation for showing opinion on public opinion channel: a case study," in Baba, N., Jain, L.C., and Howlett, R.J. (eds.), *Knowledge-Based Intel-*

ligent Information Engineering Systems & Allied Technologies(KES'2001), part 1, pp. 344-347, IOS Press, Amsterdam.

Cartwright, D.P. and Zander, A.F. (1968), *Group Dynamics: Research and Theory*, Harper and Row, New York, 3rd edition.

Duncker, K. (1945), "On problem solving," *Psychological Monographs*, vol. 58, pp. 1-112.

Flavell, J.H. (1987), "Speculations about the nature and development of metacognition," in Weinert, F.E. and Kluwe, R.H. (eds.), *Metacognition, Motivation, and Understanding*, LEA Publishers, Hillsdale NJ.

Fujihara, N. (1998), "Categorization and background knowledge," PhD Thesis, Osaka University, Osaka. [In Japanese.]

Fujihara, N. (1999), "Does Public Opinion Channel facilitate create knowledge?" *Proceedings of the 14th Annual Conference of Japanese Society for Artificial Intelligence*, pp. 103-105. [In Japanese.]

Fujihara, N. (2000), "Dynamic knowledge interaction in human cognition," in Nishida, T. (ed.), *Dynamic Knowledge Interaction*, Chap. 3, CRC Press, Florida.

Fujihara, N. and Miura, A. (2000), "The effect of the nature of a task on the strategy to search information from Internet," *XXVII International Congress of Psychology*. (Abstract available in *International Journal of Psychology*, vol. 35, no. 3/4, p. 84.)

Fukuhara, T., Takeda, H., and Nishida, T. (2000), "Multiple-text summarization for collective knowledge formation," in Nishida, T. (ed.), *Dynamic Knowledge Interaction*, Chap. 3, CRC Press, Florida.

Gick, M.L. and Holyoak, K.J. (1980), "Analogical problem solving," *Cognitive Psychology*, vol. 12, pp. 306-355.

Guilford, J.P. (1961), "Factorial angles of psychology," *Psychological Review*, vol. 68, pp. 1-10.

Kawakita, J. (1967), *Method of Thinking*, Chuko Shinsyo, Tokyo. [In Japanese.]

Kubota, H., Nishida, T., and Koda, T. (2000), "Exchanging tacit community knowledge by talking-virtualized-egos," *Proceedings of Fourth International Conference on Autonomous Agents (Agents 2000)*, pp. 285-292. Available online at http://www.kc.t.u-tokyo.ac.jp/~kubota/.

Kurohashi, S. and Nagao, M. (1994), "KN Parser: Japanese dependency/case structure analyzer," *Proceedings of the International Workshop on Sharable Natural Language Resources*, pp. 48-55, Nara Institute of Science and Technology.

Lea, M., O'Shea, T., Fung, P., and Spears, R. (1994), " 'Flaming' in computer-mediated communication: observations, explanations, implications," in Lea, M. (ed.), *Contexts of Computer-Mediated Communication*, Harvester Wheatsheaf, London, pp.89-112.

Medin, D.L., Wattenmaker, W.D., and Hampson, S.E. (1987), "Family resemblance, conceptual cohesiveness, and category construction," *Cognitive Psychology*, vol. 19, pp. 242-279.

Newcomb, T.M. (1953), "An approach to the study of communicative acts," *Psychological Review*, vol. 60, pp. 393-404.

Nishida, T., Fujihara, N., Azechi, S., Sumi, K., and Hirata, T. (1999), "Public Opinion Channel for communities in the information age," *New Generation Computing*, vol. 17, pp. 417-427.

Nishida, T. (ed.) (2000), *Dynamic Knowledge Interaction*, CRC Press, Florida.

Nonaka, I. and Takeuchi, H. (1995), *The knowledge-creating company: how Japanese companies create the dynamics of innovation*, Oxford University Press, New York.

Osborn, A.F. (1953), *Applied Imagination*, Scribner's, New York.

Sannomiya, M. (1995), "Use of discussion for communication training to promote metacognition: implication from practical course for educational technology as preservice guidance for student teachers," *Bulletin of Research Center for School Education*, vol. 9, pp. 53-61, Naruto University of Education. [In Japanese.]

Sherif, M., Harvey, O.J., White, B.J., and Sherif, C.W. (1966), "Intergroup conflict and cooperation: the robber's cave experiment," Institute of Group Relations, University of Oklahoma, Norman OK, USA.

Appendix

Summary of the interviews

S1 (28-year-old male computer scientist)

- I feel uneasy when I get anonymous messages.

S2 (31-year-old male cognitive psychologist)

- I don't bother to use the prototype POC system continuously because the information and opinions available on it are not of interest to me.
- Although the system is anonymous, I try to infer the identity of the person who sent a message.
- Having to use a mouse and keyboard is annoying. The system should use another input method, such as speech recognition.

S3 (31-year-old female secretary)

- I don't want to read unorganized messages.
- Searching for what I need at the time is important to me.

S4 (26-year-old male computer scientist)

- I found some interesting messages on the POC, but there are also many junk messages.
- It seems that POC Communicator is an instant message system. I can certainly use it for sending short messages.
- When I find responses to my messages, I am happy and want to reply to them soon.

S5 (31-year-old female social psychologist)

- I got new knowledge from messages I had not been interested in.
- The indicated message sequence contains many loose topics. It is not appropriate for a discussion of one subject.

S6 (30-year-old male social psychologist)

- I use the POC system to record my thoughts when thinking aloud.
- Some events, such as awarding a prize to the user who writes the 100th message, would be encouraged users to write messages.

Chapter 8

A New Era of Intelligent e-Commerce Based on Intelligent Java Agent-Based Development Environment (iJADE)

R.S.T. Lee

With the rapid growth of e-commerce applications, Internet shopping is becoming part of our daily lives. Traditional Web-based product searching based on keywords searching seems insufficient and inefficient in the "sea" of information. In this chapter, the author proposes an innovative intelligent multi-agent based environment, namely (iJADE) – intelligent Java Agent Development Environment – to provide an integrated and intelligent agent-based platform in the e-commerce environment. In addition to contemporary agent development platforms, which focus on the autonomy and mobility of the multi-agents, iJADE provides an intelligent layer (also known as the "conscious layer") to implement various AI functionalities in order to produce "smart" agents.

From the implementation point of view, this chapter introduces two typical intelligent e-commerce applications using the iJADE framework, namely (a) iJADE Authenticator: an invariant face recognition intelligent mobile agent system which can provide a fully automatic, mobile and reliable user authentication service; and (b) iJADE WShopper: an innovative intelligent agent-based solution in MEB (Mobile Electronic Business) with the integration of various contemporary Web and AI technologies including WAP technology for the implementation of mobile e-commerce application with the integration of fuzzy-neural networks as the AI backbone – an extension of the previous research on fuzzy agent-based shopping using FShopper technology.

1 Introduction

Owing to the rapid development of e-commerce, ranging from C2C e-commerce applications such as e-auction to sophisticated B2B e-commerce activities such as e-Supply Chain Management (eSCM), the Internet is becoming a common virtual marketplace for us to do business, search for information and communicate with one another. However, owing to the ever-increasing amounts of information in cyberspace, information searching, or more precisely, knowledge discovery and Web-mining, is becoming the critical key to success for doing business in the cyberworld. With the advance of PC computing technology in terms of computational speed and popularity, intelligent software applications known as agents, with their distinguishing features such as autonomous properties, automatic delegation of jobs, and highly mobile and adaptive behavior in the Internet environment, are becoming a potential area of development for intelligent e-business (*i*eB) (Chan *et al.* 2001) in the new millennium (Klusch 1999).

In a typical e-shopping scenario, there are two fundamental aspects of functionality in which Web-mining and visual data mining might help. The first is customer authentication. Traditional authentication, based on username and password over a security transport layer such as the SSL (Secure Socket Layer) protocol, although providing a secured user authentication scheme, requires the customer's pro-active login in order to grant access, which may discourage the customer from his or her shopping intention. Other authentication schemes based on digital certificates with smart card technology (Rankl and Effing 1997), or biometric authentication techniques based on iris or palm recognition, might provide an alternative automatic authentication scheme. However, they all need special authentication equipment which limits usability in the e-commerce environment, not to mention raising the legal implications of accessing personal privacy data such as iris and palm patterns. In contrast, automatic authentication based on human face recognition overcomes all these limitations. In terms of visual processing equipment, the standard Web-camera is already good enough for facial pattern extraction, and is nowadays more or less standard equipment for Web browsing. Moreover, this kind of authentication scheme can provide a truly automatic scheme in which the customer does not need to provide any special identity information. More importantly, it does not need to explore any 'confidential or sensitive' data such as fingerprints and iris patterns.

The other area is the automation of the online shopping process via agent technology. Traditional shopping models include consumer buying behavior models such as the Blackwell (Engel and Blackwell 1982) and Howard-Sheth (1969) models, which all share a similar list of six fundamental stages of consumer buying behavior: (1) consumer requirement definition, (2) product brokering, (3) merchant brokering, (4) negotiation, (5) purchase and delivery, and (6) after-sale services and evaluation. In reality, the first three stages in the consumer buying behavior model involve a wide range of uncertainty and possibilities – or what we called 'fuzziness' – ranging from the setting of buying criteria and provision of products by the merchant, to the selection of goods. So far, these are all 'gray areas' that we need to explore thoroughly in order to apply agent technology to the e-commerce environment.

This chapter proposes an integrated intelligent agent-based framework, known as *i*JADE – **I**ntelligent **J**ava **A**gent-based **D**evelopment **E**nvironment. To accommodate the deficiency of contemporary agent software platforms such as IBM Aglets (http://www.trl .ibm.co.jp/aglets/) and ObjectSpace Voyager Agents (http://www .genmagic.com), which mainly focus on multi-agent mobility and communication, iJADE provides an ingenious layer called the 'Conscious (Intelligent) Layer', which supports different AI functionalities to multi-agent applications. From the implementation point of view, we will demonstrate two typical intelligent e-Commerce applications using the iJADE framework, namely (a) iJADE Authenticator: an invariant face recognition intelligent mobile agent system which can provide a fully automatic, mobile and reliable user authentication service; and (b) iJADE WShopper: an innovative intelligent agent-based solution in MEB (Mobile Electronic Business) with the integration of four different technologies: (1) WAP technology for mobile e-commerce (in the iJADE 'Support Layer'), (2) mobile agent technology based on aglets (in the iJADE 'Technology Layer'), (3) Java servlets for servlet-side agent dispatch in WAP servers, and (4) AI capability in the 'Conscious Layer' using fuzzy-neural networks as the AI backbone - an extension of the previous research on fuzzy agent-based shopping using FShopper technology (Lee and Liu 2000a).

This chapter is organized as follows. Section 2 presents an overview of face recognition and the contemporary work on invariant human face recognition. Section 3 gives a general description of agent systems for e-commerce applications. Section 4 presents the model framework of iJADE, and the two major components: 'iJADE Authenticator' for automatic user authentication and 'iJADE WShopper' for intelligent agent-based shopping. System implementation will be discussed in Section 5, which is followed by a brief conclusion.

2 Face Recognition – a Perspective

Among the various techniques and models used for Human Face Recognition, there are three major common approaches: the Template Matching Approach, the Pictorial Feature Approach and Non-model Based Gray Level Analysis.

2.1 Template versus Features

In the simplest version of template matching, the query image, represented as a bi-dimensional array of intensity values, is compared with these templates using a suitable metric which represents all the facial features.

A rather different and more complex approach considers the use of the Parameterized Model Template Matching technique. In this approach, a deformable template of a facial feature model is matched with the query image, and minimization of the matching energy function is applied for facial recognition (Lanitis *et al.* 1997, Yuille 1991). The deformable models are hand-constructed from parameterized curves that outline facial features such as mouth, eyebrow and facial outline. An energy function is defined that "attracts" the template model to the preprocessed query image, and model fitting is evaluated by minimizing these energy functions.

In the Pictorial Feature Approach, a pixel-based representation of facial features is matched against the query image. This representation could be templates of major facial features or the weight of hidden layer nodes in the neural networks. Correlation on preprocessed versions of the query image is the typical matching metric. These neural network approaches construct a network where implicit feature templates are "learnt" from the "training" image set.

Unlike the previous two approaches, Non-model Based Grey Level Analysis does not find features with semantic content such as eye, mouth, nose or eyebrow detector. Instead, features are defined by the local gray level structure of the images themselves, such as corners (Azarbayejani *et al.* 1992), symmetry (Reisfeld and Yeshurun 1992) or the "end-inhibition" feature vectors which are extracted from a wavelet decomposition of the facial image (Manjunath *et al.* 1992).

2.2 Invariance Aspects

The wide variations in face appearance under changes in pose, lighting, and expression make face recognition a highly complex problem. While existing systems do not allow much flexibility in pose, lighting and expression, some do provide certain flexibility by using invariant representations or performing an explicit geometrical normalization step.

In many cases, face recognition is not designed to handle changes in facial expression (e.g. gimmick faces) or rations out of the image plane. In tackling changes, pose and lighting with the invariant representations and normalization technique as described above, most of the current systems treat face recognition generally as a rigid, 2D problem. Some exceptions exist. They use multiple views (Akamatsu *et al.* 1992) and flexible matching strategies (von der Malsburg 1988, Wiskott and von der Malsburg 1995) to deal with some degree of expression and out-of-plane rotation.

2.3 Experimental Issues

The evaluation of face recognition systems is highly empirical, requiring consistent experimental studies on a set of test images. Therefore, the major aspects for consideration involve examining the correct/false recognition rate, the size of the image gallery and the speed of recognition. Studying the recognition results from different researchers, we note that some have achieved a high recognition rate using a limited number of sample images. For example, Baron (1981) achieved an impressive 100% recognition rate of 42 people and a false access rate of 0% on 108 images. Others attained an acceptable rate of recognition on sufficiently large image libraries, but spent a long time on network learning and image preprocessing. For instance, Kruger (1997) reported challenging recognition results of an average 90% using a FERET database consisting of 350 persons with library images of 1500 in size. Nevertheless, it took 12 hours for the learning of weights for all "jet" components.

3 Mobile Agent Technology on E-commerce

The Internet is an ideal platform for supporting e-commerce. The current Web system is catalyzing the development of e-commerce over the Internet. Specifically, we call this Internet commerce. The current Internet commerce system is primarily based on a client and server architecture. Basically, all transactions are carried out by many request/response interactions over the Internet. As the Internet is a best-effort network, sometimes a user may experience a long response time. Another approach is to use a mobile agent-based system. This involves sending a mobile software agent to a remote system using various technologies (such as IBM Aglets (http://www.trl.ibm.co.jp/aglets/), ObjectSpace Voyager Agents (http://www.objectspace.com/voyager/), FTP Software Agents (http://www.ftp.com), the General Magic Odyssey Agent System (http://www.genmagic.com), and the Agent Builder Environment from IBM (http://www.networking.ibm.com/iag)), so that the agent can conduct multiple interactions with the software resident on the remote system. The output of the interactions is then sent back to the user. An agent can also interact with other agents via the Internet before returning to the original system. It is expected that this type of agent-based system will complement the existing client/server-based Internet commerce system by providing a more advanced service.

Currently, there are many different e-commerce systems around the world, ranging from simple online shops to more complex systems that provide different types of services. Some examples include:

- BargainFinder – a database search engine for searching online music stores (http://bf.cstar.ac.com/bf).
- AuctionBot – a generic auction server that allows suppliers to auction products (http://auction.eecs.umich.edu).
- MAGNET – a system for networked electronic trading (Dasgupta *et al.* 1999).

Although concurrent agent-based systems provide an effective framework for the dispatching, communication and management of multi-mobile agents in the Internet environment, in the 'intelligent agent' there is a lack of support for the intelligent functionality in the systems.

For instance, IBM Aglets (http://www.trl.ibm.co.jp/aglets/) provide comprehensive mobile agent application interfaces (APIs), ranging from creating, dispatching, cloning, retracting and disposing aglets using AgletContext class, to their messaging and collaboration using Aglet Message classes and Mobility adapters. However, intelligent capabilities such as frame-based learning and data mining on the macroscopic level or neural-network modeling, fuzzy and genetic learning on the microscopic level have not been adopted, let alone with the development of 'truly' intelligent agent applications.

In addition, typical agent development platforms such as IBM Aglets rely heavily on the agent 'kernel', namely the Tahiti server, for maintenance, management and provision of memory spaces (namely Agent Context) for the collaboration of mobile agents. In other words, for all the client machines and backend servers (e.g. Web servers) which need to dispatch and handle agents, a dedicated agent management application (e.g. Tahiti server) must be installed beforehand. In contemporary PC technology and capability, the installation of these 'housekeeping' applications does not pose a problem. However, in a mobile e-commerce situation, if we want to make use of agent technology using WAP phones with tiny memory capacity and limited communication speed, installation of these agent management applications before invoking any mobile agents is totally infeasible and impractical under contemporary WAP technology.

4 iJADE Architecture

4.1 iJADE Framework: ACTS Model

In this chapter, we propose a fully integrated intelligent agent model called iJADE (pronounced 'IJ') for intelligent agent-based e-commerce applications. The system framework is shown in Figure 1.

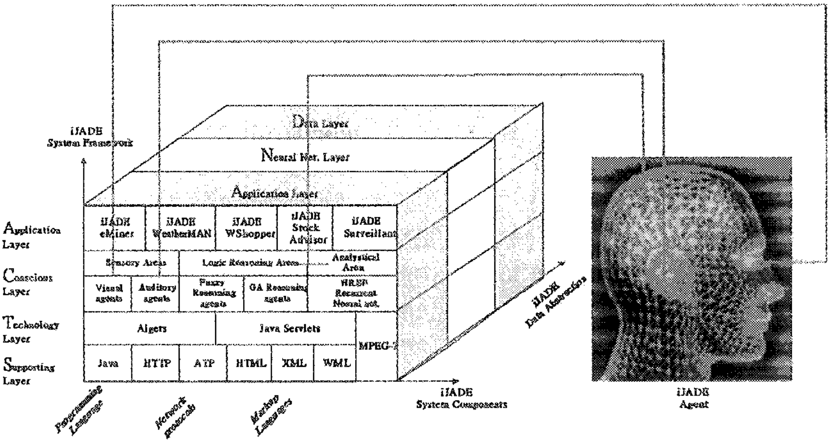


Figure 1. System Architecture of iJADE (v1.6) model.

Unlike contemporary agent systems and APIs such as IBM Aglets (http://www.trl.ibm.co.jp/aglets/) and ObjectSpace Voyager (http://www.objectspace.com/voyager/), which focus on multi-agent communication and autonomous operations, the aim of iJADE is to provide comprehensive 'intelligent' agent-based APIs and applications for future e-commerce and Web-mining applications.

Figure 1 depicts the two-level abstraction in the *i*JADE system: (a) *i*JADE system level – **ACTS** model, and (b) *i*JADE data level – **DNA** model. The ACTS model consists of (1) the **A**pplication Layer, (2) the **C**onscious (Intelligent) Layer, (3) the **T**echnology Layer, and (4) the **S**upporting Layer. The DNA model is composed of the **D**ata Layer, the **N**eural Network Layer, and the **A**pplication Layer.

Compared with contemporary agent systems which provide minimal and elementary data management schemes, the *i*JADE DNA model provides a comprehensive data manipulation framework based on neural network technology. The 'Data Layer' corresponds to the raw data and input 'stimulates' (such as the facial images captured from the Web camera and the product information in the cyberstore) from the environment. The 'Neural Network Layer' provides the 'clustering' of different types of neural networks for the purpose of 'organizing', 'interpreting', 'analyzing' and 'forecasting' operations based on the inputs from the 'Data Layer', which are used by the *i*JADE applications in the 'Application Layer'.

Another innovative feature of the *i*JADE system is the ACTS mode, which provides a comprehensive layering architecture for the implementation of intelligent agent systems, and will be explained in the following sections.

4.2 Application Layer of iJADE Model

This is the uppermost layer, which consists of different intelligent agent-based applications. These iJADE applications are developed by the integration of intelligent agent components from the 'Conscious Layer' and the data 'knowledge fields' from the DNA model.

Concurrent applications (iJADE v1.6) implemented in this layer include:

- iJADE Stock Advisor (Lee and Liu 2001b), an intelligent agent-based stock prediction system using a time series neuro-oscillatory prediction technique (Lee and Liu 2000c).
- iJADE eMiner (Lee and Liu 2001a), the intelligent Web-mining agent system on e-shopping.
- iJADE WeatherMAN (Lee and Liu 2001c), an intelligent weather forecasting agent which is the extension of previous research on multi-station weather forecasting using fuzzy neural networks (Liu and Lee 1999). Unlike traditional Web-mining agents, which focus on the automatic extraction and provision of the latest weather information, iJADE WeatherMAN possesses neural network-based weather forecasting capability (AI services provided by the 'Conscious Layer' of the iJADE model) to act as a 'virtual' weather reporter as well as an 'intelligent' weather forecaster for weather prediction.
- iJADE WShopper (Lee 2001), an integrated intelligent fuzzy shopping agent with WAP technology for intelligent mobile shopping on the Internet (Lee and Liu 2000a).
- iJADE Authenticator, the automatic agent-based invariant face recognition system for user authentication presented in this chapter.

4.3 Conscious (Intelligent) Layer

This layer provides the intelligent basis of the iJADE system, using the agent components provided by the 'Technology Layer'. The 'Conscious Layer' consists of the following three main intelligent functional areas:

1. Sensory Area – for the recognition and interpretation of incoming stimulates. It includes (a) visual sensory agents using the EGDLM (Elastic Graph Dynamic Link Model) for invariant visual object recognition (Lee and Liu 1999a,b,c), and (b) auditory

sensory agents based on the wavelet-based feature extraction and interpretation technique (Hossain *et al.* 1999).

2. Logic Reasoning Area – conscious area providing different AI tools for logical 'thinking' and rule-based reasoning, such as fuzzy and GA (Genetic Algorithms) rule-based systems (Lee and Liu 2000b).
3. Analytical Area – consists of various AI tools for analytical calculation, such as recurrent neural network-based analysis for real-time prediction and data mining (Lee and Liu 2000c).

4.4 Technology Layer Using IBM Aglets and Java Servlets

This layer provides all the necessary mobile agent implementation APIs for the development of intelligent agent components in the 'Conscious Layer'.

In the current version (v1.6) of the iJADE model, IBM Aglets (http://www.trl.ibm.co.jp/aglets/) are used as the agent 'backbone'. The basic functionality and runtime properties of aglets are defined by the Java Aglet, AgletProxy and AgletContext classes. The abstract class aglet defines the fundamental methods that control the mobility and lifecycle of an aglet. It also provides access to the inherent attributes of an aglet, such as creation time, owner, codebase and trust level, as well as dynamic attributes, such as the arrival time at a site and the address of the current context.

The main function of the AgletProxy class is to provide a handle that is used to access the aglet. It also provides location transparency by forwarding requests to remote hosts and returning results to the local host. Actually, all communication with an aglet occurs through its aglet proxy. The AgletContext class provides the runtime execution environment for aglets within the Tahiti server. Thus, when an aglet is dispatched to a remote site, it is detached from the current AgletContext object, serialized into a message bytestream, sent across the network, and reconstructed in a new AgletContext, which in turn provides the execution environment at the remote site. The other critical component of the Aglet environment is the security issue. Aglets provide a security model in the form of an AgletSecurityManager, which is a subclass of the "standard" Java SecurityManager.

In this layer, server-side computing using Java Servlet technology is also adopted due to the fact that for certain intelligent agent-based applications, such as the WShopper (Lee 2001), in which limited resources (in terms of memory and computational speed) are provided by the WAP devices (e.g. WAP phones), all the iJADE agents interactions are invoked in the 'backend' WAP server using Java Servlet technology.

4.5 Supporting Layer

This layer provides all the necessary system support to the 'Technology Layer'. It includes (1) Programming language support based on Java; (2) Network protocol support such as HTTP, HTTPS, ATP, and so on; and (3) Markup language support such as HTML, XML, WML, and so on.

5 Implementation

In this chapter, two major iJADE applications used to support intelligent e-Commerce are presented, namely 1) iJADE Authenticator: to support intelligent user authentication based on the invariant face recognition technique, and 2) iJADE WShopper: to support intelligent agent-based shopping using the fuzzy-neuro technique.

5.1 iJADE Face Recognizer: System Overview

*i*JADE Authenticator mainly consists of two subsystems, one at the client (i.e. customer) site and the other at the server (e.g. virtual shopping mall) site. A schematic diagram of the whole system is shown in Figure 2.

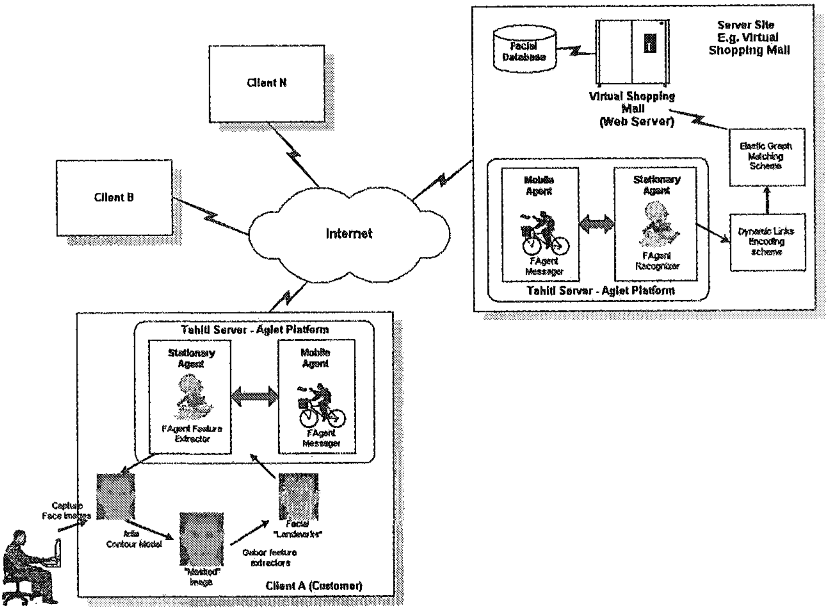


Figure 2. Schematic diagram of iJADE Authenticator.

In summary, there are three kinds of intelligent agents operating within the system. They are:

1. FAgent Feature Extractor – A stationary agent situated within the client machine to extract the facial features from the facial image which is captured by the client's digital camera.
2. FAgent Messager – A mobile agent who acts as a messager that on the one hand "carries" the facial features to the server-side agent and on the other hand "reports" the latest status back to the client machine.

3. FAgent Recognizer – A stationary agent situated within the server (e.g. virtual shopping mall). Its main duty is to perform invariant facial pattern matching against the server-side facial database.

5.1.1 Client-Side Subsystem

Basically, the client-side subsystem consists of the following three stage operations:

1. Facial image capturing stage using the client's desktop video camera.
2. Facial contour extraction stage using Active Contour Model (ACM).
3. Automatic facial landmarks extraction stage using Gabor feature extractor.

Facial contour extraction stage – Active Contour Model (ACM)

The Active Contour Model (Blake and Isard 1998) involves the use of a 'snake' (Kass *et al.* 1987) to locate the face contour. The 'snake' is a continuous curve that forms an initial state (facial template) and tries to deform itself dynamically on the image picture. This is a result of the action of external forces that attract the snake towards image features and internal forces which maintain the smoothness of the template's shape (Figure 3). The sum of the membrane energy, denoting the snake stretching, and the thin-plate energy, denoting the snake bending, gives the following snake energy:

$$E_{\text{int}}(u(s))=\alpha(s)\left|u_s(s)\right|^2+\beta(s)\left|u_{ss}(s)\right|^2 \tag{1}$$

where $u(s) = (x(s), y(s))$ is the snake curve and s is the arc-length of the curve. The parameters of elasticity α and β control the smoothness of the snake curve.

The deformation of the "snake" is governed by external forces. These forces are associated with a potential *P(x,y)* which, in general, is defined in terms of the gradient module of the image convoluted by a Gaussian function:

$$P(x,y) = -\left|\nabla(G(x,y) * I(x,y))\ \right| \qquad (2)$$

or as a distance map of the edge points:

$$P(s,y)=d(x,y), \quad P(x,y)=-e^{-d(x,y)^2} \qquad (3)$$

where *d(x,y)* denotes the distance between the pixel *(x,y)* and its closest edge point. The snake is moved by potential forces and tries to fall in a valley as if it were under the effect of gravity.

The total snake energy is given by the functional energies sum as:

$$E_{snake} = \int_0^1 E_{\text{int}} + E_{ext} ds = \int_0^1 \alpha(s)\left|u_s(s)\right|^2 + \beta(s)\left|u_{ss}(s)\right|^2 + P(u(s))ds \quad (4)$$

The minimum of the snake energy satisfies an Euler-Lagrange equation:

$$-\frac{d}{ds}(\alpha u_s(s)) + \frac{d^2}{ds^2}(\beta u_{ss}(s)) + \nabla P(u(s)) = 0 \qquad (5)$$

and boundary conditions.

Automatic facial landmarks extraction scheme

In this module, according to the 50 facial landmarks (e.g. nose, eyes, eye-brows, mouth, facial contours, etc.) defined in the "deformed" facial template (Figure 4), Gabor filters of 15 different frequency bands (ϕ) and 8 different orientations (θ) are used. A total of 120 feature vectors of different attributes are extracted automatically from these landmark positions. The filter function is given as follows:

$$g_{\phi,\theta}(x,y)=\frac{1}{\sigma\sqrt{\pi}}e^{(-\frac{x^2+y^2}{2\sigma^2})}e^{2\pi i\phi(x\cos\theta+y\sin\theta)} \tag{6}$$

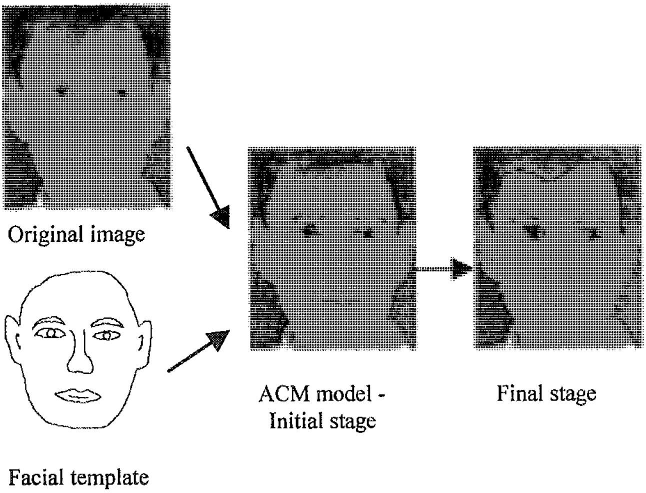


Figure 3. Facial contour extraction using ACM.

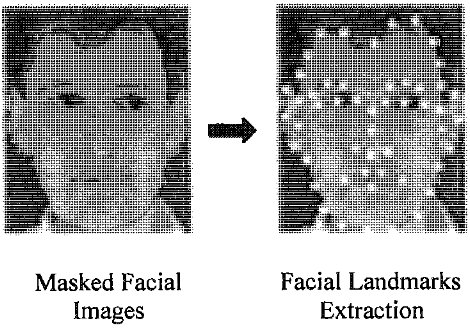


Figure 4. Facial features extraction scheme.

5.1.2 Server-Side Subsystem

The server-side subsystem basically consists of the following modules:

- Dynamic Links Initialization scheme.
- Elastic attribute graph matching scheme between the query image and the images from the facial database.

In the Dynamic Link Initialization process, dynamic links ($z_{ij,kl}$) between "memory" facial attribute graphs and figure objects from the images gallery are initialized according to the following rules:

$$z_{ij,kl} = \varepsilon J_{ij} J_{kl} \quad \text{for } J_{ij} \in A,\ J_{kl} \in B, \tag{7}$$

where Js are the feature vectors extracted from the facial landmarks and ε is the parameter value between 0 and 1; A and B denote the figure and memory graphs respectively.

In the Elastic Graph Matching Module (Lee and Liu 1999b), the attribute graph of the figure is "dynamically" matched with each "memory" object attribute graph by minimizing the energy function $H(z)$:

$$H(z) = -\sum_{i,j\in B;k,l\in A} z_{ij} z_{jl} z_{ik} z_{kl} + \gamma \sum_{i\in B}\left(\sum_{k\in A} z_{ik} - 1\right)^2 + \gamma \sum_{k\in A}\left(\sum_{i\in B} z_{ik} - 1\right)^2 \tag{8}$$

within tolerance level μ.

$H(z)$ is minimized using the gradient descent:

$$z_{ij}(t+1) = \left[z_{ij}(t) - \eta \frac{\partial H(z(t))}{\partial z_{ij(t)}} \right]^{w}, \tag{9}$$

where $[\ldots]^w$ denotes the value of z_{ij} confined to the interval $[0,w]$. At equilibrium (within a chosen tolerance level μ), $H(z)$ will be minimized, and the connection pattern in the memory layer represents the pattern recalled by the figure pattern.

5.2 iJADE WShopper: a Fuzzy-neuro-Based E-Shopping System

As an extension to the previous work on Fuzzy Shopper (FShopper) (Lee and Liu 2000a), a fuzzy shopping agent for Internet shopping, *i*JADE WShopper provides an integrated intelligent agent-based solution for m-shopping via a WAP device. Based on the *i*JADE model discussed in Section 4, *i*JADE WShopper integrates the following technologies to develop the application: 1) Mobile agent technology based on Aglets for the agent framework (the 'Technology Layer' of the *i*JADE model), 2) Java Servlets technology for the manipulation of the server-side operations in the brokering machine (the 'Technology Layer' of the *i*JADE model), and 3) FShopper - intelligent fuzzy-neural based shopping operations (the 'Conscious Layer' of the *i*JADE model).

Figure 5 depicts the overall system framework of *i*JADE WShopper on m-shopping (mobile shopping via WAP phone) using *i*JADE technology in different cyberstores. Actually, Figure 5 demonstrates two situations of 'intelligent agent shopping': 1) Fuzzy Internet shopping via a Web browser, and 2) Fuzzy WAP shopping (WShopper) using a WAP phone as the WAP device. In other words, any agent-based cyberstores can be operated in this framework provided that their agent servers conform to MASIF (Mobile Agent System Interoperability Facility) standards. More importantly, under this infrastructure, both Web-based e-shopping and MEB m-shopping can operate simultaneously!

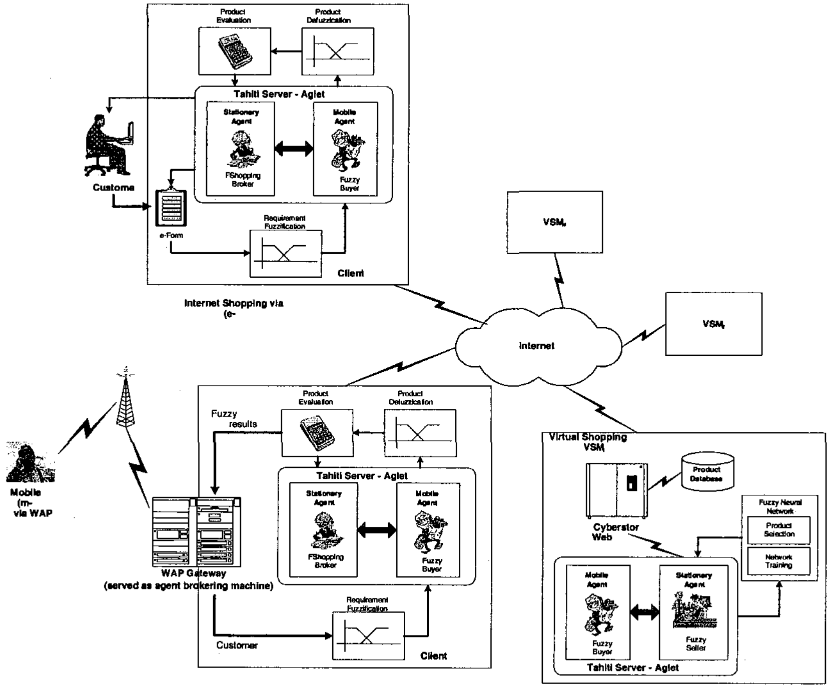


Figure 5. System overview of *i*JADE IWShopper for mobile shopping.

The system framework of the IWShopper consists of the following modules:

- Customer requirement definition (CRD)
- Requirement fuzzification scheme (RFS)
- Fuzzy agents negotiation scheme (FANS)
- Fuzzy product selection scheme (FPSS)
- Product defuzzification scheme (PDS)
- Product evaluation scheme (PES)

In this *i*JADE agents brokering center, there are two types of *i*JADE agents: (1) FShopping Broker – A stationary agent that acts as a buyer broker on behalf of the customer. This autonomous *i*JADE agent contains all the necessary information and analytical techniques (provided by the 'Conscious Layer' of the model), such as the requirements for fuzzification and defuzzication, and product evaluation techniques; (2) Fuzzy Buyer – A mobile *i*JADE agent that acts as a virtual buyer in the virtual marketplace. This corresponds to all agent communication, interaction and negotiation operations.

After the customer has input all his/her product requirements (e.g. color, size, style, fitness) into the WAP phone, WShopping Broker (in the brokering center) will convert all these fuzzy requirements into fuzzy variables by using the "embedded" knowledge (i.e. the membership functions) with its knowledge base. Of course, WShopping Broker will also be responsible for the form data validation jobs as well. Sample fuzzy membership functions for selected attributes for shoes, including color and degree of fitness, are shown in Figure 6.

$$Color\,(color) = \{red, yellow, blue\}$$

$$\underline{color} = \begin{Bmatrix} Light, Normal, Deep \mid \forall \mu_{\underline{color}}(Light), \\ \mu_{\underline{color}}(Normal), \mu_{\underline{color}}(Deep) \in [0,1] \end{Bmatrix}$$

$$\mu_{\underline{color}}(Light) = \begin{cases} 1 & if\ 0 \le x \le 64 \\ \dfrac{96-x}{32} & if\ 64 \le x \le 96 \\ 0 & otherwise \end{cases}$$

$$\mu_{\underline{color}}(Normal) = \begin{cases} \dfrac{x-64}{32} & if\ 64 \le x \le 96 \\ 1 & if\ 96 \le x \le 160 \\ \dfrac{192-x}{32} & if\ 160 \le x \le 192 \\ 0 & otherwise \end{cases}$$

$$\mu_{\underline{color}}(Deep) = \begin{cases} 1 & if\ x \ge 192 \\ \dfrac{x-160}{32} & if\ 160 \le x \le 192 \\ 0 & otherwise \end{cases}$$

$$\underline{Fitness} = \begin{Bmatrix} Loose, Fit, Tight \mid \forall \mu_{\underline{fitness}}(Loose), \\ \mu_{\underline{fitness}}(Fit), \mu_{\underline{fitness}}(Tight) \in [0,1] \end{Bmatrix}$$

$$\mu_{\underline{fitness}}(Loose) = \begin{cases} 1 & if\ 0 \le x \le 3 \\ 4-x & if\ 3 \le x \le 4 \\ 0 & otherwise \end{cases}$$

$$\mu_{\underline{fitness}}(Fit) = \begin{cases} x-3 & if\ 3 \le x \le 4 \\ 1 & if\ 4 \le x \le 7 \\ 8-x & if\ 7 \le x \le 8 \\ 0 & otherwise \end{cases}$$

$$\mu_{\underline{fitness}}(Tight) = \begin{cases} 1 & if\ x \ge 8 \\ x-7 & if\ 7 \le x \le 8 \\ 0 & otherwise \end{cases}$$

Figure 6. Sample Membership Functions for Color and Degree of Fitness.

Once the Fuzzy Seller has collected all the customer fuzzy requirements, it will perform the product selection based on a fuzzy neural network (provided by the *i*JADE DNA data model). Actually, the fuzzy neural network is an integration of fuzzy technology and the Feedforward Backpropagation neural network (FFBP) provided by the *i*JADE Conscious Layer. A schematic diagram of the network framework is depicted in Figure 7.

Figure 7 illustrates the FPSS using a fuzzy-neural network for product selection (e.g. a pair of shoes). The fuzzy neural network consists of two parts: the fuzzy module and the FeedForward BackPropropagation (FFBP) neural network module. The fuzzy module provides the network with a bundle of fuzzy variables as input nodes. In the example, the fuzzy variables consist of color components (i.e. red, yellow and blue), size, length, degree of fitness and price. (Detail explanations of Fuzzy Theory, Neural Networks and the system framework for the applications of Fuzzy-neural network based on Feed-forward Backpropagation (FFBP) model can be found in (Lee and Liu 2000b)).

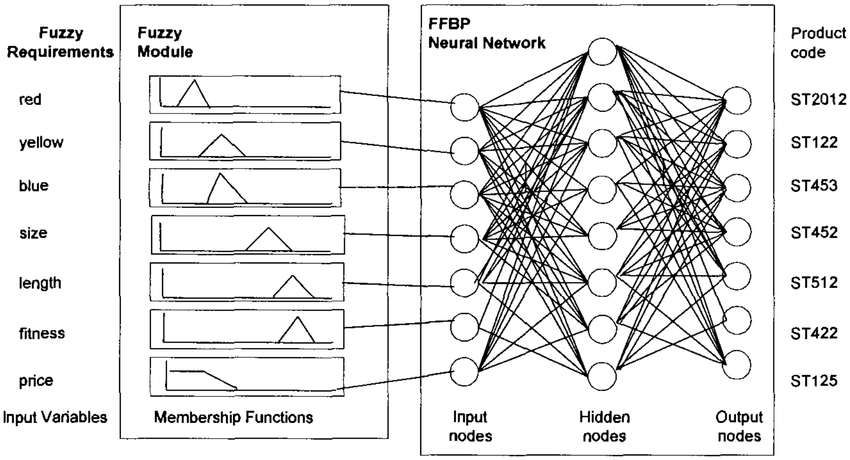


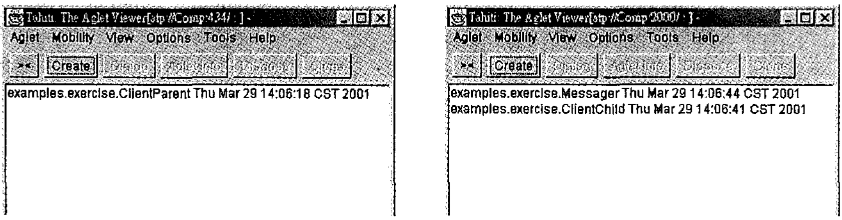
Figure 7. Fuzzy-neural network for product selection.

6 Experimental Results

6.1 iJADE Authenticator

In the experiment, 100 human subjects were used for system training. A set of 1,020 tested patterns resulting from different facial expressions, viewing perspectives, and sizes of stored templates were used for testing. A series of tested facial patterns was obtained with a CCD camera providing a standard video signal, and digitized at 512×384 pixels with 8 bits of resolution.

The computer system that we adopted to implement and measure the performance of the hybrid system was a SUN-Sparc 20 workstation. Sample iJADE agents' activities screens and snapshots of authentication screens are shown in Figures 8 and 9.


Tahiti: The Aglet Viewer[atp://Comp:434/ :] -
Aglet Mobility View Options Tools Help
>< Create
Tahiti: The Aglet Viewer[atp://Comp:2000/ :] -
Aglet Mobility View Options Tools Help
>< Create

```
examples.exercise.ClientParent Thu Mar 29 14:06:18 CST 2001
```

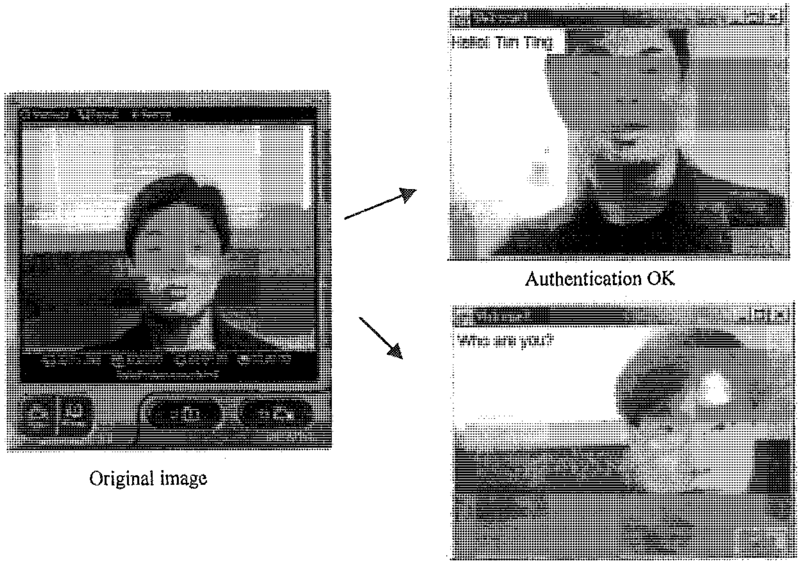
```
examples.exercise.Messager Thu Mar 29 14:06:44 CST 2001
examples.exercise.ClientChild Thu Mar 29 14:06:41 CST 2001
```

Figure 8. iJADE agents' activities.

6.1.1 iJADE Authenticator Test I: Viewing Perspective Test

In this test, a viewing perspective ranging from –30° to +30° (with reference to the horizontal and vertical axis) was adopted, using 100 test patterns for each viewing perspective. The recognition results are presented in Table 1.

According to the "Rotation Invariant" property of the EGDLM model (Lee and Liu 1999b,c), the FAgent possesses the same characteristic in the "contour maps elastic graph matching" process. An overall correct recognition rate of over 86% was achieved.



Hello! Tim Ting

Who are you?

Original image

Authentication OK

Authentication fail

Figure 9. Authentication screens.

Table 1. Results of viewing perspective test.

Viewing perspectives (from horiz. axis)	Correct classification	Viewing perspectives (from vertical axis)	Correct classification
+30°	84%	+30°	86%
+20°	90%	+20°	88%
+10°	92%	+10°	91%
–10°	91%	–10°	92%
–20°	89%	–20°	87%
–30°	85%	–30°	82%

6.1.2 iJADE Authenticator Test II: Facial Pattern Occlusion and Distortion Test

In this test, the 120 test patterns are basically divided into three categories:

- Wearing spectacles or other accessories
- Partial occlusion of the face by obstacles such as cups / books (in reading and drinking processes)
- Various facial expressions (such as laughing, angry and gimmicky faces).

Pattern recognition results are shown in Table 2.

Table 2. Recognition results for occlusion/distortion test.

Pattern Occlusion & Distortion Test	Correct classification
Wearing spectacles (or other accessories)	87%
Face partially hidden by obstacles (e.g. books, cups)	72%
Facial expressions (e.g. laughing, angry and gimmicky faces)	83%

Compared with the three different categories of facial occlusion, "wearing spectacles" has the least negative effect on facial recognition, owing to the fact that all the main facial contours are still preserved in this situation. In the second situation, the effect on the recognition rate depends on what proportion and which portion of the face is obscured. Nevertheless, the average correct recognition rate was found to be over 73%.

Facial expressions and gimmicky faces gave the most striking results. Owing to the "Elastic Graph" characteristic of the model, the recognition engine "inherited" the "Distortion Invariant" property and an overall correct recognition rate of 83% was attained.

6.2 *i*JADE WShopper

From the implementation point of view, m-shopping in cyberstores is performed for simulation purposes. For the product database, over 200 items under eight categories were used to construct the e-catalog. These categories were: T-shirt, shirt, shoes, trousers, skirt, sweater, tablecloth, napkins. We deliberately chose softgood items instead of hardgoods such as books or music (as commonly found in most e-shopping agent systems), so that it would allow more room for fuzzy user requirement definition and product selection. Figure 10 depicts the sample screen shots of intelligent mobile shopping using WAP simulators as illustration.

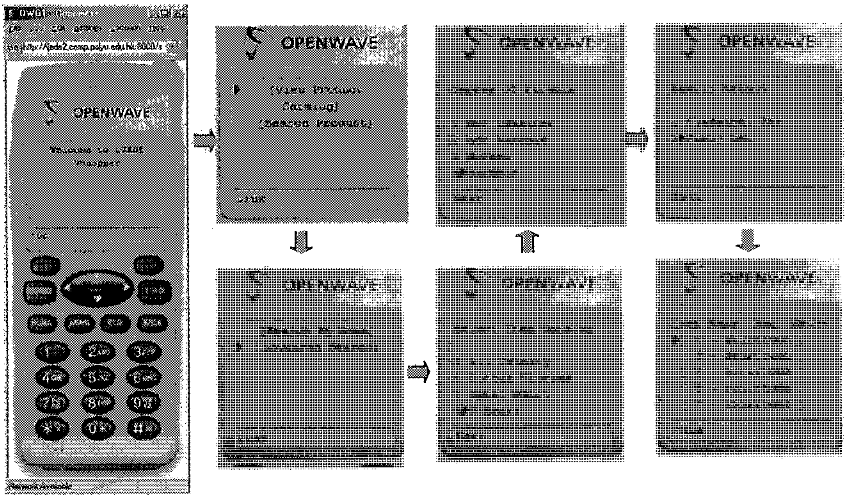


Figure 10. Sample screen shots of WShopper using WAP simulator.

For neural network training, all the e-catalog items were 'pre-trained' in the sense that we had pre-defined the attribute descriptions for all these items to be 'fed' into the fuzzy neural network for product training (for each category). Totally, eight different neural networks were constructed according to each different category of product.

From the experimental point of view, two sets of tests were conducted: the Round Trip Time (RTT) test and the Product Selection (PS) test. The RTT test aims at evaluating the "efficiency" of the WShopper in the sense that it will calculate the whole round trip time of the iJADE agents, instead of calculating the difference between the arrival and departure time to/from any particular server. The RTT test will calculate all the 'component' time fragments starting from the collection of the user requirement from the WAP phone, through fuzzification, to the product selection and evaluation steps in the brokering center (WAP gateway) and various cyberstores, so that a total picture of the performance efficiency can be deduced. A comparison with the fuzzy e-shopper (FShopper (Lee and Liu 2000a)) will be conducted.

In the Product Selection (PS) test, since there was no definite answer to whether a product would 'fit' the taste of the customer or not, a sample group of 40 candidates was used to judge the 'effectiveness' of the WShopper. Details are given in the following sections.

6.2.1 Round Trip Time (RTT) Test

In this test, two iJADE Servers were used: the T1server and the T2server. The T1server was situated within the same LAN as the client machine, while the T2server was located in a remote site (on campus).

The results of the mean RTT after 100 trials for each server are shown in Table 3.

As shown in Table 3, the total RTT is dominated by the Fuzzy Product Selection Scheme (FPSS), but the time spent is still within an acceptable timeframe: 5 to 7 seconds. Further, the differences of RTT between the servers situated in the same LAN and those at the remote sites were not significant except in the FANS, where the Fuzzy Buyer needed to take a slightly longer 'trip' than the others. Of course, in reality, this factor depends heavily on the network traffic.

Table 3. Mean RTT summary after 100 trials.

Time (msec.)	WShopper (m-shopping)		FShopper (e-shopping)	
	T1server	T2sever	T1server	T2sever
Server location	Same LAN as client	Remote site (on campus)	Same LAN as client	Remote site (on campus)
A. In WAP phone & WAP gateway (WShopper) / Client browser (FShopper)				
CRD	-		-	
RFS	25	73	310	305
B. In Cyberstore (both WShopper & FShopper)				
FANS	225	1304	320	2015
FPSS	3120	3311	4260	4133
A. In WAP phone & WAP gateway (WShopper) / Client browser (FShopper)				
PDS	310	335	320	330
PES	53	102	251	223
TOTAL RTT	3733	5125	5461	7006

Compared with e-shopping using the FShopper (Lee and Liu 2000a), m-shopping using the WShopper provides a more efficient result, for two main reasons: 1) In the WShopper scenario, all the cyberstores and WAP gateways are configured with the iJADE agent framework, better management of fuzzy shopping operations is provided, and more importantly the fuzzy agents are 'light-weighted' since all the related fuzzy evaluation APIs are implicitly provided by the iJADE framework. 2) As the major task of the WAP device is the collection of customer requirements and the display of selection results, all the invoking, dispatching and manipulation work of fuzzy agents (which was originally done in the client machine) is now switched to the brokering center, as reflected by the short processing time in the RFS and PES processes.

6.2.2 Product Selection (PS) Test

Unlike the RTT test, in which objective figures can be easily obtained, the PS test results rely heavily on user preference. In order to achieve a more objective result, a sample group of 40 candidates was invited for system evaluation. In the test, each candidate would "buy" one product from each category according to his/her own requirements. For evaluation, they would browse around the e-catalog to choose a list of the 'best five choices' (L) which 'fit their taste'. In comparison with the 'top five' recommended product items (i) given by the fuzzy shopper, the 'Fitness Value (FV)" is calculated as follows:

$$FV = \frac{\sum_{n=1}^{5} n \times i}{15} \quad where\ i = \begin{cases} 1 & if\ i \in L \\ 0 & otherwise \end{cases} \tag{10}$$

In the calculation, scores of 5 to 1 were given to 'correct matches' of the candidate's first to fifth 'best five' choices with the fuzzy shopper's suggestion. For example, if out of the five "best choices" selected by the customer, products of rank nos. 1, 2, 3 and 5 appear in the fuzzy shopper recommended list, the fitness value will be 73%, which is the sum of 1, 2, 3 and 5 divided by 15.

In this experiment, four different product selection schemes are adopted:

- Simple product selection (using product description matching – traditional technique)
- Product selection based on FFBP neural network training
- Product selection based on fuzzy product description – no network training is involved
- WShopper – product selection based on fuzzy-neural training

The corresponding Fitness Values (FV) and the degree of improvement (against the 'traditional' technique) under the eight different product categories are shown in Table 4.

In view of the WShopper PS result, it is not difficult to predict that the performance of the Fuzzy Shopper is highly dependent on the "variability" (or "fuzziness") of the merchandise. The higher the fuzziness (which means the greater the variety), the lower the score. As shown in Table 4, skirts and shoes are typical examples in which the skirts category scores 65% and shoes 89%. Nevertheless, the average score is still over 81%. Note that these figures are only for illustration purposes, as human justification and product variety in actual scenarios do vary case by case.

Table 4. Fitness values for the eight different product categories under different product selection schemes.

Product category	Fitness Value FV% (% Improvement)			
	Simple product search	Pure FFBP network training	Pure fuzzy product search	WShopper (fuzzy-neural)
T-shirt	48	58 (+21)	54 (+13)	81 (+ 69)
Shirt	41	48 (+17)	53 (+29)	78 (+ 90)
Shoes	56	68 (+21)	58 (+ 4)	89 (+ 59)
Trousers	52	63 (+21)	56 (+ 8)	88 (+ 69)
Skirts	32	38 (+19)	45 (+41)	65 (+103)
Sweater	45	53 (+18)	55 (+22)	81 (+ 80)
Tablecloth	57	67 (+18)	61 (+ 7)	85 (+ 49)
Napkins	53	64 (+21)	58 (+ 9)	86 (+ 62)
Average score	48.0	57.4 (+20)	55.0 (+15)	81.6 (+ 70)

Comparing different product selection techniques, the WShopper outperforms the FShopper by over 40%; compared with the 'traditional technique', a promising improvement of 48% is attained.

Another interesting phenomenon is found when comparing the PS of the 'Pure FFBP training' with the 'Pure Fuzzy PS'. Overall, although the former outperforms the latter by 5%, the 'Pure Fuzzy PS' technique produces 'exceptionally good' results in certain product categories such as shirt, skirts and sweater, which are all 'fuzzy' products in which the fuzzification technique might help in fuzzyproduct selection.

7 Conclusion

In this chapter, an innovative intelligent agent-based system framework – the iJADE model – is proposed to facilitate the implementation of intelligent agent-based e-Commerce applications. From the implementation point of view, two typical intelligent agent-based e-Commerce applications, namely iJADE Authenticator and iJADE Wshopper, are introduced as illustration. This development will hopefully signal a new era of e-Commerce applications using intelligent agent-based systems.

Acknowledgment

The author is grateful for departmental financial support for the *i*JADE projects, including iJADE Framework (v2.0) (Z042) and iJADE WShopper (Z028), from the Hong Kong Polytechnic University.

References

Akamatsu, S., Sasaki, T., Fukamachi, H., Masui, N., and Suenaga, Y. (1992), "An accurate and robust face identification scheme," *Proceedings International Conference on Pattern Recognition*, vol. 2, The Hague, The Netherlands, pp. 217-220.

Azarbayejani, A., Starner, T., Horowitz, B., and Pentland, A. (1992), "Visually Controlled Graphs," Technical Report 180, MIT Media Lab., Vision Modeling Group.

Baron, R.J. (1981), "Mechanisms of human facial recognition," *Int. Journal of Man Machine Studies*, pp. 137-178.

Blake, A. and Isard, M. (1998), *Active Contours*, Springer.

Chan, H.C.B., Lee, R.S.T., Dillion, T.S., and Chang, E. (2001), *E-Commerce: Fundamentals and Applications*, John Wiley and Sons Ltd.

Dasgupta, P., Narasimhan, N., Moser, L.E., and Smith, P.M. (1999), "MAGNET – mobile agents for networked electronic trading," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 4, pp. 509-525.

Engel, J. and Blackwell, R. (1982), *Consumer Behavior*, CBS College Publishing.

Hossain, I., Liu, J., and Lee, R. (1999), "A study of multilingual speech feature: perspective scalogram based on wavelet analysis," *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol. II, pp. 178-183, Tokyo, Japan.

Howard, J. and Sheth, J. (1969), *The Theory of Buyer Behavior*, John Wiley and Sons.

Kass, M., Witkin, A., and Terzopoulos, D. (1987), "Snakes: active contour models," *Proceedings International Conference on Computer Vision*, pp. 259-268.

Klusch, M. (ed.) (1999), *Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet*, Springer-Verlag Berlin Heidelberg.

Kruger, N. (1997), "An algorithm for the learning of weights in discrimination functions using prior constant," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 764-768.

Lanitis, A., Taylor, C.J., and Cootes, T.F. (1997), "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 743-756.

Lee, R.S.T. and Liu, J.N.K. (1999a), "An automatic satellite interpretation of tropical cyclone patterns using elastic graph dynamic link model," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, no. 8, pp. 1251-1270.

Lee, R.S.T. and Liu, J.N.K. (1999b), "An integrated elastic contour fitting and attribute graph matching model for automatic face coding and recognition," *Proceedings of the Third International Conference on Knowledge-Based Intelligent Information Engineering Systems (KES'99)*, Adelaide, Australia, IEEE Press, pp. 292-295.

Lee, R.S.T. and Liu, J.N.K. (1999c), "An oscillatory elastic graph matching model for scene analysis," *Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'99)*, Las Vegas, Nevada, USA, pp. 42-45.

Lee, R.S.T. and Liu, J.N.K. (2000a), "Fuzzy Shopper - a fuzzy network based shopping agent in E-commerce environment," *Proc. of the International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-commerce (MAMA 2000)*, December 11-13, Wollongong, Australia.

Lee, R.S.T. and Liu, J.N.K. (2000b), "Teaching and learning the A.I. modeling," in Jain, L.C. (ed.), *Innovative Teaching Tools: Knowledge-Based Paradigms*, (*Studies in Fuzziness and Soft Computing* **36**), Physica-Verlag, Springer, pp. 31-86.

Lee, R.S.T. and Liu, J.N.K. (2000c), "Tropical cyclone identification and tracking system using integrated neural oscillatory elastic graph matching and hybrid RBF network track mining techniques," *IEEE Transaction on Neural Networks*, vol. 11, no. 3, pp. 680-689.

Lee, R.S.T. (2001), "iJADE IWShopper – a new age of intelligent mobile Web shopping system based on fuzzy-neuro agent technology," *Web Intelligence: Research and Development*, (*LNAI* 2198), Springer-Verlag, pp. 403-412.

Lee, R.S.T. and Liu J.N.K. (2001a), "iJADE eMiner – a Web-based mining agent based on intelligent Java agent development environment (iJADE) on Internet shopping," *Advances in Knowledge Discovery and Data Mining*, (*Lecture Notes in Artificial Intelligence series LNAI* 2035), Springer-Verlag, pp. 28-40.

Lee, R.S.T. and Liu, J.N.K. (2001b), "iJADE Stock Predictor – an intelligent multi-agent based time series stock prediction system," *Intelligent Agent Technology: Research and Development*, World Scientific, pp. 495-499.

Lee, R.S.T. and Liu, J.N.K. (2001c), "iJADE WeatherMAN – a multiagent fuzzy-neuro network based weather prediction system," *Intelligent Agent Technology: Research and Development*, World Scientific, pp. 424-433.

Liu, J.N.K. and Lee, R.S.T. (1999), "Rainfall forecasting from multiple point source using neural networks," *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol. II, Tokyo, Japan, pp. 429-434.

Manjunath, B.S., Shekhar, C., Chellappa, R., and von der Malsburg, C. (1992), "A robust method for detecting image features with application to face recognition and motion correspondence," *Proceedings International Conference on Pattern Recognition*, vol. 2, pp. 208-212.

Rankl, W. and Effing, W. (1997), *Smart Card Handbook*, Wiley.

Reisfeld, D. and Yeshurun, Y. (1992), "Robust detection of facial features by generalized symmetry," *Proceedings International Conference on Pattern Recognition*, vol. 1, The Hague, The Netherlands, pp. 117-120.

Von der Malsburg, C. (1988), "Pattern recognition by label graph matching," *Neural Networks*, vol. 1, no. 1, pp. 141-148.

Wiskott, L. and von der Malsburg, C. (1995), "Recognizing faces by dynamic link matching," *Proceedings ICANN '95*, Paris, pp. 347-352.

Yuille, A.L. (1991), "Deformable templates for face recognition," *Journal of Cognitive Neuroscience*, pp. 59-70.

This page is intentionally left blank

Chapter 9

Automated Internet Trading Based on Optimized Physics Models of Markets

L. Ingber and R.P. Mondescu

We describe a real-time, internet-based S&P futures trading system, including a description of general aspects of internet-mediated interactions with electronic exchanges. Inner-shell stochastic nonlinear dynamic models are developed, and Canonical Momenta Indicators (CMI) are derived from a fitted Lagrangian used by outer-shell trading models dependent on these indicators. Recursive and adaptive optimization using Adaptive Simulated Annealing (ASA) is used for fitting parameters shared across these shells of dynamic and trading models.

1 Introduction

Launching and exploiting a successful automated trading system implies accomplishing two major tasks, of almost equal significance:

- designing and developing a robust trading model of markets of interest,
- connecting the system to markets, addressing two problems
  - the communications hardware infrastructure,
  - the software interface.

To develop a robust and consistent model of markets, we should remark that real-world problems are rarely solved in closed algebraic form, yet methods must be devised to deal with this complexity to extract practical informations in finite time. This is indeed true in the field of financial engineering, where time series of various financial instruments reflect non-equilibrium, highly non-linear, possibly even chaotic (Peters 1991) underlying processes. A further difficulty is the huge amount of data necessary to be processed. Under these circumstances, to develop models and schemes for automated, profitable trading is a non-trivial task.

Apparently, the connectivity task involves mostly a programming effort, where a host of technical tools may considerably simplify the task. In practice an equal amount of work must be devoted to a proper design of various software components and solving multiple hardware problems, given the following constraints:

- necessity of accessing multiple markets.
- lack of a standard API (Application Programming Interface) for accessing different exchanges.
- lack of an universal language of communication between financial institutions.
- stringent reliability requirements posed on the communication infrastructure.

Currently, there are sustained efforts toward an unified, non-proprietary financial "electronic" language (FIX – Financial Information Exchange – open protocol (FIX Protocol 2000)). FIX approach is to define and promote a common set of types of messages, their format and the session-level interaction, for communicating securities transactions between two parties, in a real-time electronic trading environment.

1.1 Approaches

Detailed discussions pertinent to the theoretical model underlying the trading system and computational aspects were published previously, see (Ingber and Mondescu 2001).

Regarding the financial modeling aspect, in the context of this chapter, it is important to stress that dealing with such complex systems invariably requires modeling of dynamics, modeling of actions on these dynamics, and algorithms to fit parameters in these models to real data. We have elected to use methods of mathematical physics for our models of the dynamics, artificial intelligence (AI) heuristics for our models of trading rules acting on indicators derived from our dynamics, and methods of sampling global optimization for fitting our parameters. Too often there is confusion about how these three elements are being used for a complete system. For example, in the literature often there is discussion of neural net trading systems or genetic algorithm trading systems. However, neural net models (used for either or both models discussed here) also require some method of fitting their parameters, and genetic algorithms must have some kind of cost function or process specified to sample a parameter space, and so on.

Some powerful methods have emerged during years, appearing from at least two directions: One direction is based on inferring rules from past and current behavior of market data leading to learning-based, inductive techniques, such as neural networks, or fuzzy logic. Another direction starts from the bottom-up, trying to build physical and mathematical models based on different economic prototypes. In many ways, these two directions are complementary and a proper understanding of their main strengths and weaknesses should lead to synergetic effects beneficial to their common goals.

Among approaches in the first direction, neural networks already have won a prominent role in the financial community. This is due to their ability to handle large quantities of data and to uncover and model nonlinear functional relationships between various combinations of fundamental indicators and price data (Azoff 1994, Gately 1996).

In the second direction we can include models based on nonequilibrium statistical mechanics (Ingber 2000) fractal geometry (Mandelbrot 1997), turbulence (Mantegna and Stanley 1996), spin glasses and random matrix theory (Laloux *et al.* 1999), renormalization group (Johansen *et al.* 1999), and gauge theory (Ilinsky and Kalinin 1997). Although the very complex nonlinear multivariate character of financial markets is recognized (Hull 2000), these approaches seem to have had a lesser impact on current quantitative finance practice, although it is increasing becoming clear that this direction can lead to practical trading strategies and models.

To bridge the gap between theory and practice, as well as to afford a comparison with neural networks techniques, we focus on presenting an effective trading system of S&P futures, anchored in the physical principles of non-equilibrium statistical mechanics applied to financial markets (Ingber 1984, 2000).

Starting with nonlinear, multivariate, nonlinear stochastic differential equation descriptions of the price evolution of cash and futures indices, we build an algebraic cost function in terms of a Lagrangian. Then, a maximum likelihood fit to the data is performed using a global optimization algorithm, Adaptive Simulated Annealing (ASA) (Ingber 1993a). As firmly rooted in field theoretical concepts, we derive market canonical momenta indicators, and we use these as technical signals in a recursive ASA optimization that tunes the outer-shell of trading rules. We do not employ metaphors for these physical indicators, but rather derive them directly from models fit to data.

The outline of the chapter is as follows: Just below we briefly discuss the optimization method and momenta indicators.

In Section 2 we discuss some general, technical elements related to building an internet-based interface between the provider of financial services (e.g., an exchange) and the client using an electronic trading system.

In the ensuing two sections we establish the theoretical framework supporting our model, and the statistical mechanics approach together with the optimization method, respectively. In Section 5 we detail the trading system, and in Section 6 we describe our results. Our conclusions are presented in Section 7.

1.2 Optimization

Large-scale, non-linear fits of stochastic nonlinear forms to financial data require methods robust enough across data sets. (Just one day, tick data for regular trading hours could reach 10,000-30,000 data points.) Simple regression techniques exhibit deficiencies with respect to obtaining reasonable fits. They too often get trapped in local minima typically found in nonlinear stochastic models of such data. ASA is a global optimization algorithm that has the advantage – with respect to other global optimization methods as genetic algorithms, combinatorial optimization, and so on – not only to be efficient in its importance-sampling search strategy, but to have the statistical guarantee of finding the best optima (Ingber 1989, Ingber and Rosen 1993). This gives some confidence that a global minimum can be found, of course provided care is taken as necessary to tune the algorithm (Ingber 1996a).

It should be noted that such powerful sampling algorithms also are often required by other models of complex systems than those we use here (Ingber 1993b). For example, neural network models have taken advantage of ASA (Cohen 1994, Cozzio-Buëler 1995, Indiveri *et al.* 1993), as have other financial and economic studies (Mayer *et al.* 1996, Sakata and White 1998).

1.3 Indicators

In general, neural network approaches attempt classification and identification of patterns, or try forecasting patterns and future evolution of financial time series. Statistical mechanical methods attempt to find dynamic indicators derived from physical models based on general principles of non-equilibrium stochastic processes that reflect certain market factors. These indicators are used subsequently to generate trading signals or to try forecasting upcoming data.

In this chapter, the main indicators are called Canonical Momenta Indicators (CMI), as they faithfully mathematically carry the significance of market momentum, where the "mass" is inversely proportional to the price volatility (the "masses" are just the elements of the metric tensor in this Lagrangian formalism) and the "velocity" is the rate of price changes.

The concept of momentum is at least intuitively appreciated by all traders. Many traders use some algorithm to calculate the momenta of markets they are trading, e.g., perhaps to use as supplemental indicators to confirm other indicators to act on trades.

Markets increasingly are becoming inter-dependent, effectively defining a larger collective multivariate market. Many traders account for such circumstances by at least following indicators of other markets in addition to those they are explicitly trading. Clearly, it would be beneficial to have accurate measures of such inter-dependencies, beyond statistical correlations, to have indicators that measure the importance of inter-dependencies of the dynamic evolution of the markets. However, it also would be useful if such information could be presented in an understandable intuitive manner, without altering any detailed content. Canonical momenta can satisfy this wish-list, and a detailed application to trading is described below.

2 Connection to Electronic Exchanges

The growth of internet as a communication infrastructure and the exponential increase in computer power drastically altered the mechanics of securities trading. Electronic matching of orders eliminates market makers and brokers as intermediaries, allowing a vast increase in the number of market participants and better terms for financial execution of trading orders.

Despite more or less visible obstructions by the traditional players, electronic exchanges appeared or traditional exchanges converted to electronic ones (DTB – Germany, Matif – France, LIFFE – UK, Eurex – Germany and Switzerland merged futures exchanges) and their volume exploded (Burghardt 2001).

Intra-day price feeds, real-time streaming quotes (even order books – commonly referred to as Level II quotes – (Archipelago 2001)) and integrated trade systems are available at almost no cost, and complicated models could be programmed and run by all market participants.

Sophisticated automated systems at large financial institutions could browse a wealth of data and filtered it, based on various theoretical models, in the search of the arbitrage opportunity.

All these developments have made more prominent the role and the functionality of the interface connecting the trading system to the provider of financial services (which include both data sources and exchanges). By *financial services* we refer throughout to services related to trading (submission of orders, trading support or clearing services) provided by an exchange or other financial institutions to an end user client.

As a software application, a trading system has mainly two components: the computational kernel and the connection API. We talk here about the connection API at the client organization level. The API is the software layer allowing a trading tool of the client, the trader, to communicate with the software of the exchange or other provider of financial services.

Based on the data (prices, volume, time, various indicators) input and on the theoretical model used, the computational kernel generates the trading signals and sends them to the order execution module, a component of the connection API.

The connection API must address two classes of problems:

1. Access to real-time price quotes.
2. Execution of the trade order.

We remark that above and in what follows we choose to use – for clarity purposes – the term *connection API* as a rather broad grouping of functional units that may not necessarily reflect a more constrained software engineering point of view. For example, in most cases the data access component requires a separate, independent development effort from the order execution module.

A more complex, commercial version of a connection API should have certain features, among which we list

- enables universal access to multiple exchanges with unique API,
- allows proprietary trading tools or other systems to connect to the order execution system,
- provides compatibility with multiple financial instruments (stocks, bonds, futures, and so on),
- provides order routing service with real-time updates and various execution types and order qualifiers,
- provides back-office services (trade confirmations, profit/loss reports, execution reports, full order book update, settlement prices),
- provides market news services (market opening/closing announcements, market updates, instruments status/specifications changes),
- provides queries service: range of trades, range of prices, product specification changes.

Collecting and processing real-time price data could be done using 3rd party applications (two random examples: Reuters Triarch real-time services, ESignal data services (eSignal 2001)), or by directly writing into the API provided by the exchange, e.g., the Chicago Mercantile Exchange (CME) Market Data API – MDAPI 1.0 – or the Eurex Values/Gate 3.0 API (Eurex 2001).

Usually, most vendors provide integrated solutions, essentially trading applications that combine both the data and the execution systems. These applications are usually black-box systems that does not offer a lower level control of data, trading signals and trading orders, imperative requirements for building a proprietary trading tool.

We focus next on describing the technological and design aspects common to the connection API, with emphasis on the order routing component of the API. We choose to do so because it is more complex than the data access module and less details are available to a general audience.

2.1 Internet Connectivity: Overview

In general, connecting a trading system directly to one (or multiple) exchanges is a process requiring support and control from the dedicated technology and marketing departments of the exchange. It is reasonably understood that the trading system cannot be launched live without passing several quality control check-points, imposed both by in-house and exchange Quality Assurance (QA) departments.

The evolution of the trading application from concept to production tool could be subscribed to the following milestones:

- initial software development (concept, design, proto-type),
- advanced development,
- technical certification with sub-stages
  - functional testing,

  - failover/recovery testing,
  - stress testing,
- network certification,
- pre-production testing
  - connectivity testing,
  - clearing cycle (end-to-end) testing.

Associated with these development stages, various requirements (hardware and software) must be met within the automated trading environment. We describe these requirements below.

2.2 Internet Connectivity: Hardware Requirements

Reliable data feeds are critical components of a successful automated trading system. Internet access to exchanges through 3rd party applications/intermediaries and standard communication infrastructure (modems, cable modems, DSL, and so on) is possible, but due to reliability concerns and higher probability of connection breakdowns, it is limited for trading systems operating at longer time scales (daily, weekly trades) and lower trading volumes, or to personal trading.

When trading time scale decreases to minutes or seconds and large transactions, direct access to exchanges, with dedicated lines is required.

For both data access and order routing, the development, initial testing and certification phases require at least an ISDN line. The production stage necessitates frame relay (e.g., 256k AT&T) and ISDN connections as main communication backbone, and back-up lines, respectively.

Routers (e.g., Cisco 800, 2610) and possibly, a separate diagnostic line, are also required, as well as some 3rd party software applications (e.g., Reuters TIBCO).

All this equipment is usually installed by exchange personnel in collaboration hardware manufacturers technical support. Costs and timelines for hardware deployment should be factored in when evaluating capabilities of a trading model.

2.3 Internet Connectivity: Software Requirements

Besides design aspects, important considerations are the choice of language and development platform. At this moment, preponderantly for trading engines requiring fast execution, Java still does not offer the required speed and reliability. The languages of choice remain C++ and C.

Although at the client level, the computational kernel could be developed on any software platform, the need to interface with the API provided by exchanges limits considerably the platform choices: currently, Windows NT and Sun Solaris are the preferred operating systems, with some exchanges supporting also IBM AIX.

Moreover, commercial development environments (as Microsoft Visual Studio or Sun Workshop) and sometimes 3rd party libraries (e.g., Rogue Wave (RogueWave 2001)) are also necessary (at least when reaching certification and production levels), as only these are usually supported by exchanges.

2.4 API Order Execution Module: Components and Functionality

In terms of design, the connection API must insulate the computational kernel of various code changes operated by outside providers (e.g., exchanges) to which the system is connected. Function of specific interests, various design patterns (factory, template, bridge, façade, adapter (Gamma *et al.* 1994)) could be applied.

The basic order of events necessary to be handled by the order routing and execution component of the connection API is:

1. initialization (instantiate various object factories, register with the server to receive responses, and so on),
2. connect to exchange API server (open session),
3. authenticate connection (login),
4. subscribe to a particular instrument (or multiple instruments), or to a particular field of a instrument (e.g., bid prices for a certain stock),
5. create and submit orders,
6. terminate communication with the exchange server and disconnect.

After opening the trading session, the connection API should insure (when queried) that connection status and execution reports are available.

Various types of order (market order, stop order, limit order, stop limit order, market if touched = the opposite of a stop order) and types of time-in-force (we list here only those suitable for automated trading) must be handled by the order routing module. The particular order type and time-in-force type applied in actual trading are chosen function of the characteristics of the trading model:

- fill-or-kill, a limit order, which is canceled if not filled immediately and completely,
- fill-and-kill, a limit order that, if not filled completely, all remaining quantity is cancelled,
- good-till-cancel, an order to be held until filled or until is cancelled.

Note that not all of these above qualifiers are necessarily supported by the exchange of interest.

The main task of the order execution API is to create orders. An order will contain several fields, among which we list the most important:

- order identification number,

- exchange identification code,
- instrument identifier,
- order type (market, limit, stop,...),
- execution type (fill-and-kill, and so on),
- price (for stop, limit, stop-limit orders),
- quantity,
- time of entry.

The order execution API component sends and receives (generally FIX-compliant) messages. We quote several of them below:

- single order (new order for a single instrument),
- cancel request (request to cancel an order),
- cancel/replace request (a request to cancel a previous order and replace it with a new order),
- status request (a request for status of an order),
- heartbeat (a periodic signal send by exchange server to verify that connection is alive),
- reject (the order was rejected by the exchange server),
- cancel reject (the cancel request send by the client was rejected by the exchange server),
- execution report.

Logic for taking appropriate action function of the message (or combination of messages) received must be implemented at the API level, in connection with signals produced by the computational engine.

Finally, from a development point of view, correct processing of previous categories of messages is essential. In particular some points need attention:

- the cancel/replace logic, which may depend on the exchange (e.g., with the CME FIX API the client needs to send a status request to check the state of an order),
- the closing of a session (should be done gracefully, otherwise lost messages or damaged session accounting could occur),

- error handling (all possible errors/exceptions should be dealt properly),
- connection management (a crucial component of a connection API. The API should dynamically monitor and react to connectivity problems).

3 Models

3.1 Langevin Equations for Random Walks

The use of Brownian motion as a model for financial systems is generally attributed to Bachelier (Bachelier 1900), though he incorrectly intuited that the noise scaled linearly instead of as the square root relative to the random log-price variable. Einstein is generally credited with using the correct mathematical description in a larger physical context of statistical systems. However, several studies imply that changing prices of many markets do not follow a random walk, that they may have long-term dependences in price correlations, and that they may not be efficient in quickly arbitraging new information (Jensen 1978, Mandelbrot 1971, Taylor 1982). A random walk for returns, rate of change of prices over prices, is described by a Langevin equation with simple additive noise η, typically representing the continual random influx of information into the market.

$$\begin{gathered} \dot{M} = -f + g\eta , \\ \dot{M} = \frac{dM}{dt} , \\ < \eta(t) >_\eta = 0, \quad < \eta(t), \eta(t') >_\eta = \delta(t - t'), \end{gathered} \tag{1}$$

where f and g are constants, and M is the logarithm of (scaled) price, $M(t) = \log(P(t)/P(t - dt))$. Price, although the most dramatic observable, may not be the only appropriate dependent variable or order parameter for the system of markets (Brown *et al.* 1983). This possibility has also been called the "semi-strong form of the efficient market hypothesis" (Jensen 1978).

The generalization of this approach to include multivariate nonlinear non-equilibrium markets led to a model of statistical mechanics of financial markets (SMFM) (Ingber 1984).

3.2 Adaptive Optimization of F^x Models

Our S&P model for the evolution of futures price F is

$$\begin{aligned} dF &= \mu dt + \sigma F^x dz, \\ < dz > &= 0, \\ < dz(t)\, dz(t') > &= dt \delta(t - t'), \end{aligned} \tag{2}$$

where the exponent x of F is one of the dynamical parameters to be fit to futures data together with μ and σ.

We have used this model in several ways to fit the distribution's volatility defined in terms of a scale and an exponent of the independent variable (Ingber 2000).

A major component of our trading system is the use of adaptive optimization, essentially constantly retuning the parameters of our dynamic model each time new data is encountered in our training, testing and real-time applications. The parameters $\{\mu, \sigma\}$ are constantly tuned using a quasi-local simplex code (Barabino *et al.* 1980, Nelder and Mead 1964) included with the ASA (Adaptive Simulated Annealing) code (Ingber 1993a).

We have tested several quasi-local codes for this kind of trading problem, versus using robust ASA adaptive optimizations, and the faster quasi-local codes seem to work quite well for adaptive updates after a zeroth order parameters set is found by ASA (Ingber 1996b,c).

4 Statistical Mechanics of Financial Markets (SMFM)

4.1 Statistical Mechanics of Large Systems

Aggregation problems in nonlinear nonequilibrium systems typically are "solved" (accommodated) by having new entities/languages developed at these disparate scales in order to efficiently pass information back and forth between scales. This is quite different from the nature of quasi-equilibrium quasi-linear systems, where thermodynamic or cybernetic approaches are possible. These thermodynamic approaches typically fail for nonequilibrium nonlinear systems.

Many systems are aptly modeled in terms of multivariate differential rate-equations, known as Langevin equations (Haken 1983),

$$\begin{aligned} \dot{M}^G &= f^G + \hat{g}^G_j \eta^j, (G = 1, \ldots, \Lambda)(j = 1, \ldots, N), \\ \dot{M}^G &= \frac{dM^G}{dt}, \\ < \eta^j(t) >_\eta = 0, &\quad < \eta^j(t), \eta^{j'}(t') >_\eta = \delta^{jj'} \delta(t - t'), \end{aligned} \tag{3}$$

where f^G and $\hat{g}^G_j$ are generally nonlinear functions of mesoscopic order parameters M^G, j is an index indicating the source of fluctuations, and $N \geq \Lambda$. The Einstein convention of summing over repeated indices is used. Vertical bars on an index, e.g., $|j|$, imply no sum is to be taken on repeated indices. The "microscopic" index j relates to the typical physical nature of fluctuations in such statistical mechanical systems, wherein the variables η are considered to be aggregated from finer scales relative to the "mesoscopic" variables M.

Via a somewhat lengthy, albeit instructive calculation, outlined in several other papers (Ingber 1984, 1991, Ingber *et al.*1991), involving an intermediate derivation of a corresponding Fokker-Planck or

Schrödinger-type equation for the conditional probability distribution $P[M(t)|M(t_0)]$, the Langevin rate Eq. (3) is developed into the more useful probability distribution for M^G at long-time macroscopic time event $t_{u+1} = (u+1)\theta + t_0$, in terms of a Stratonovich path-integral over mesoscopic Gaussian conditional probabilities (Cheng 1972, Dekker 1979, Graham 1978, Langouche *et al.*1979, 1980). Here, macroscopic variables are defined as the long-time limit of the evolving mesoscopic system.

The corresponding Schrödinger-type equation is (Graham 1978, Langouche *et al.* 1979)

$$
\begin{aligned}
\frac{\partial P}{\partial t} &= \frac{1}{2}(g^{GG'}P)_{,GG'} - (g^G P)_{,G} + V, \\
g^{GG'} &= \delta^{jk}\hat{g}_j^G \hat{g}_k^{G'}, \\
g^G &= f^G + \frac{1}{2}\delta^{jk}\hat{g}_j^{G'}\hat{g}_{k,G'}^G, \\
[\ldots]_{,G} &= \frac{\partial[\ldots]}{\partial M^G}.
\end{aligned}
\tag{4}
$$

This is properly referred to as a Fokker-Planck equation when $V \equiv 0$. Note that although the partial differential Eq. (4) contains information regarding M^G as in the stochastic differential Eq. (3), all references to j have been properly averaged over. I.e., $\hat{g}_j^G$ in Eq. (3) is an entity with parameters in both microscopic and mesoscopic spaces, but M is a purely mesoscopic variable, and this is more clearly reflected in Eq. (4). In the following, we often drop superscripts on M for clarity, with the understanding that M represents the vector $\{M^G\}$.

The calculation of the long-time evolution of these distributions most often defies any algebraic solution, and special techniques must be utilized. This is required, for example, to calculate many kinds of financial instruments, e.g., bond prices, options, derivatives, and so on. People have developed numerical algorithms for each representation, i.e., for the Langevin, Fokker-Planck and the Lagrangian probability representations. Methods to treat the latter are developed around the path-integral formalism:

The path integral representation can be written in terms of the prepoint discretized Lagrangian L, further discussed below (Graham 1978, Langouche *et al.* 1980, 1982),

$$
\begin{aligned}
P[M,t|M,t_0]dM(t) &= \int \dots \int \underline{D}M \exp(-S) \\
&\quad \times \delta[M(t_0)]\delta[M(t)], \\
S &= \min \int_{t_0}^{t} dt' L, \\
\underline{D}M &= \lim_{u \to \infty} \prod_{v=1}^{u+1} g^{1/2} \prod_{G} (2\pi\theta)^{-1/2} dM^G(t_v), \\
L(\dot{M}^G, M^G, t) &= \frac{1}{2}(\dot{M}^G - g^G) g_{GG'} (\dot{M}^{G'} - g^{G'}) \\
&\quad - V, \\
g_{GG'} &= (g^{GG'})^{-1}, \\
g &= \det(g_{GG'}).
\end{aligned}
\tag{5}
$$

Mesoscopic variables have been defined as M^G in the Langevin and Fokker-Planck representations, in terms of their development from the microscopic system labeled by j. The entity $g_{GG'}$, is a bona fide metric of this space (Graham 1978). Short-time "forecast" of data points is realized using the most probable path equation (Dekker 1980)

$$
\frac{dM^G}{dt} = g^G - g^{1/2}(g^{-1/2} g^{GG'})_{,G'}. \tag{6}
$$

In the literature on economics, there appears to be sentiment to define Eq. (3) by the Itô, rather than the Stratonovich prescription. It is true that Itô integrals have Martingale properties not possessed by Stratonovich integrals (Oksendal 1998) which leads to risk-neural theorems for markets (Harrison and Kreps 1979, Pliska 1997), but the nature of the proper mathematics – actually a simple transformation between these two discretizations – should eventually be determined by proper aggregation of relatively microscopic models of markets. It should be noted that virtually all investigations of other physical systems, which are also continuous time models of discrete processes, conclude that the Stratonovich interpretation coincides with reality, when multiplicative noise with zero correlation time, modeled in terms of white noise η^j, is properly considered as the limit of real noise with finite correlation time (Gardiner 1983). The path integral succinctly demonstrates the difference between the two: The Itô prescription corresponds to the prepoint discretization of L, wherein $\theta\dot{M}(t) \rightarrow M(t_{v+1}) - M(t_v)$ and $M(t) \rightarrow M(t_v)$. The Stratonovich prescription corresponds to the midpoint discretization of L, wherein $\theta\dot{M}(t) \rightarrow M(t_{v+1}) - M(t_v)$ and $M(t) \rightarrow \frac{1}{2}(M(t_{v+1}) + M(t_v))$. In terms of the functions appearing in the Fokker-Planck Eq. (4), the Itô prescription of the prepoint discretized Lagrangian L, Eq. (5), is relatively simple, albeit deceptively so because of its nonstandard calculus. In the absence of a nonphenomenological microscopic theory, the difference between a Itô prescription and a Stratonovich prescription is simply a transformed drift (Langouche *et al.* 1982).

There are several other advantages to Eq. (5) over Eq. (3). Extrema and most probable states of M^G, $\ll M^G \gg$, are simply derived by a variational principle, similar to conditions sought in previous studies (Merton 1973). In the Stratonovich prescription, necessary, albeit not sufficient, conditions are given by

$$\begin{aligned} \delta_G L &= L_{,G} - L_{,\dot{G}:t} = 0, \\ L_{,\dot{G}:t} &= L_{,\dot{G}G'}\dot{M}^{G'} + L_{,\dot{G}\dot{G}'}\ddot{M}^{G'}. \end{aligned} \tag{7}$$

For stationary states, $\dot{M}^G = 0$, and $\partial\bar{L}/\partial\bar{M}^G = 0$ defines $\ll \bar{M}^G \gg$, where the bars identify stationary variables; in this case, the macroscopic variables are equal to their mesoscopic counterparts.

Note that $\bar{L}$ is not the stationary solution of the system, e.g., to Eq. (4) with $\partial P/\partial t = 0$. However, in some cases (Ingber 1985), $\bar{L}$ is a definite aid to finding such stationary states. Many times only properties of stationary states are examined, but here a temporal dependence is included. E.g., the $\dot{M}^G$ terms in L permit steady states and their fluctuations to be investigated in a nonequilibrium context. Note that Eq. (7) must be derived from the path integral, Eq. (5), which is at least one reason to justify its development.

4.2 Algebraic Complexity Yields Simple Intuitive Results

It must be emphasized that the output of this formalism is not confined to complex algebraic forms or tables of numbers. Because L possesses a variational principle, sets of contour graphs, at different long-time epochs of the path-integral of P over its variables at all intermediate times, give a visually intuitive and accurate decision-aid to view the dynamic evolution of the scenario. For example, this Lagrangian approach permits a quantitative assessment of concepts usually only loosely defined.

$$\text{“Momentum''} = \Pi^G = \frac{\partial L}{\partial(\partial M^G/\partial t)}, \tag{8a}$$

$$\text{“Mass''} = g_{GG'} = \frac{\partial^2 L}{\partial(\partial M^G/\partial t)\partial(\partial M^{G'}/\partial t)}, \tag{8b}$$

$$\text{“Force''} = \frac{\partial L}{\partial M^G}, \tag{8c}$$

$$\text{“F = ma''} : \delta L = 0 = \frac{\partial L}{\partial M^G} - \frac{\partial}{\partial t}\frac{\partial L}{\partial(\partial M^G/\partial t)}, \tag{8d}$$

where M^G are the variables and L is the Lagrangian. These physical entities provide another form of intuitive, but quantitatively precise, presentation of these analyses. For example, daily newspapers use some of this terminology to discuss the movement of security prices.

In this chapter, the Π^G serve as canonical momenta indicators (CMI) for these systems.

4.2.1 Derived Canonical Momenta Indicators (CMI)

The extreme sensitivity of the CMI gives rapid feedback on changes in trends as well as the volatility of markets, and therefore are good indicators to use for trading rules (Ingber 1996b). A time-locked moving average provides manageable indicators for trading signals. This current project uses such CMI developed as a byproduct of the ASA fits described below.

4.2.2 Intuitive Value of CMI

In the context of other invariant measures, the CMI transform covariantly under Riemannian transformations, but are more sensitive measures of activity than other invariants such as the energy density, effectively the square of the CMI, or the information which also effectively is in terms of the square of the CMI (essentially integrals over quantities proportional to the energy times a factor of an exponential including the energy as an argument). Neither the energy or the information give details of the components as do the CMI. In oscillatory markets the relative signs of such activity can be quite important.

The CMI present single indicators for each member of a set of correlated markets, "orthogonal" in the defined metric space. Each indicator is a dynamic weighting of short-time differenced deviations from drifts (trends) divided by covariances (risks). Thus the CMI also give information complementary to just trends or standard deviations separately.

4.3 Correlations

In this chapter we report results of our one-variable trading model. However, it is straightforward to include multi-variable trading models in our approach, and we have done this, for example, with coupled cash and futures S&P markets.

Correlations between variables are modeled explicitly in the Lagrangian as a parameter usually designated ρ. This section uses a simple two-factor model to develop the correspondence between the correlation ρ in the Lagrangian and that among the commonly written Wiener distribution dz.

Consider coupled stochastic differential equations for futures F and cash C:

$$dF = f^F(F,C)dt + \hat{g}^F(F,C)\sigma_F dz_F, \tag{9a}$$

$$dC = f^C(F,C)dt + \hat{g}^C(F,C)\sigma_C dz_C, \tag{9b}$$

$$< dz_i > = 0,\ i = \{F,C\}, \tag{9c}$$

$$< dz_i(t)dz_j(t') > = dt\delta(t-t'), i = j, \tag{9d}$$

$$< dz_i(t)dz_j(t') > = \rho dt\delta(t-t'), i \neq j, \tag{9e}$$

where $< . >$ denotes expectations with respect to the multivariate distribution.

These can be rewritten as Langevin equations (in the Itô prepoint discretization)

$$\frac{dF}{dt} = f^F + \hat{g}^F\sigma_F(\gamma^+\eta_1 + \mathrm{sgn}\rho\,\gamma^-\eta_2), \tag{10a}$$

$$\frac{dC}{dt} = g^C + \hat{g}^C\sigma_C(\mathrm{sgn}\rho\,\gamma^-\eta_1 + \gamma^+\eta_2), \tag{10b}$$

$$\gamma^{\pm} = \frac{1}{\sqrt{2}}[1 \pm (1-\rho^2)^{1/2}]^{1/2}, \tag{10c}$$

$$n_i = (dt)^{1/2}p_i, \tag{10d}$$

where p_1 and p_2 are independent [0,1] Gaussian distributions.

The equivalent short-time probability distribution, P, for the above set of equations is

$$
\begin{aligned}
P &= g^{1/2}(2\pi dt)^{-1/2}\exp(-Ldt),\\
L &= \frac{1}{2}M^{\dagger}\underline{g}M,\\
M &= \begin{pmatrix} \frac{dF}{dt} - f^F \\ \frac{dC}{dt} - f^C \end{pmatrix},\\
g &= \det(\underline{g}). \qquad (11)
\end{aligned}
$$

$\underline{g}$, the metric in $\{F, C\}$-space, is the inverse of the covariance matrix,

$$
\underline{g}^{-1} = \begin{pmatrix} (\hat{g}^F\sigma_F)^2 & \rho\hat{g}^F\hat{g}^C\sigma_F\sigma_C \\ \rho\hat{g}^F\hat{g}^C\sigma_F\sigma_C & (\hat{g}^C\sigma_C)^2 \end{pmatrix}. \qquad (12)
$$

The CMI indicators are given by the formulas

$$
\Pi^F = \frac{(dF/dt - f^F)}{(\hat{g}^F\sigma_F)^2(1-\rho^2)} - \frac{\rho(dC/dt - f^C)}{\hat{g}^F\hat{g}^C\sigma_F\sigma_C(1-\rho^2)}, \qquad (13a)
$$

$$
\Pi^C = \frac{(dC/dt - f^C)}{(\hat{g}^C\sigma_C)^2(1-\rho^2)} - \frac{\rho(dF/dt - f^F)}{\hat{g}^C\hat{g}^F\sigma_C\sigma_F(1-\rho^2)}. \qquad (13b)
$$

4.4 ASA Outline

The algorithm Adaptive Simulated Annealing (ASA) fits short-time probability distributions to observed data, using a maximum likelihood technique on the Lagrangian. This algorithm has been developed to fit observed data to a theoretical cost function over a D-dimensional parameter space (Ingber 1989), adapting for varying sensitivities of parameters during the fit. The ASA code can be obtained at no charge, via WWW from http://www.ingber.com/ or via FTP from ftp.ingber.com (Ingber 1993a).

4.4.1 General Description

It helps to visualize the problems presented by such complex systems as a geographical terrain. For example, consider a mountain range, with two "parameters," e.g., along the NorthSouth and EastWest directions. We wish to find the lowest valley in this terrain. ASA approaches this problem similar to using a bouncing ball that can bounce over mountains from valley to valley. We start at a high "temperature," where the temperature is an ASA parameter that mimics the effect of a fast moving particle in a hot object like a hot molten metal, thereby permitting the ball to make very high bounces and being able to bounce over any mountain to access any valley, given enough bounces. As the temperature is made relatively colder, the ball cannot bounce so high, and it also can settle to become trapped in relatively smaller ranges of valleys.

We imagine that our mountain range is aptly described by a "cost function." We define probability distributions of the two directional parameters, called generating distributions since they generate possible valleys or states we are to explore. We define another distribution, called the acceptance distribution, which depends on the difference of cost functions of the present generated valley we are to explore and the last saved lowest valley. The acceptance distribution decides probabilistically whether to stay in a new lower valley or to bounce out of it. All the generating and acceptance distributions depend on "temperatures."

Simulated annealing (SA) was developed in 1983 to deal with highly nonlinear problems (Kirkpatrick *et al.* 1983), as an extension of a Monte-Carlo importance-sampling technique developed in 1953 for chemical physics problems. In 1984 (Geman and Geman 1984), it was established that SA possessed a proof that, by carefully controlling the rates of cooling of temperatures, it could statistically find the best minimum, e.g., the lowest valley of our example above. This was good news for people trying to solve hard problems which could not be solved by other algorithms. The bad news was that the guarantee was only good if they were willing to run SA forever. In 1987, a method of fast annealing (FA) was developed (Szu and Hartley 1987), which permitted lowering the temperature exponentially faster, thereby statistically guaranteeing that the minimum could be found in some finite time. However, that time still could be quite long. Shortly thereafter, Very Fast Simulated Reannealing (VFSR) was developed in 1987 (Ingber 1989), now called Adaptive Simulated Annealing (ASA), which is exponentially faster than FA.

ASA has been applied to many problems by many people in many disciplines (Ingber 1993b, 1996a, Wofsey 1993). The feedback of many users regularly scrutinizing the source code ensures its soundness as it becomes more flexible and powerful.

4.4.2 Multiple Local Minima

Our criteria for the global minimum of our cost function is minus the largest profit over a selected training data set (or in some cases, this value divided by the maximum drawdown). However, in many cases this may not give us the best set of parameters to find profitable trading in test sets or in real-time trading. Other considerations such as the total number of trades developed by the global minimum versus other close local minima may be relevant. For example, if the global minimum has just a few trades, while some nearby local minima (in terms of the value of the cost function) have many trades and was profitable in spite of our slippage factors, then the scenario with more trades might be more statistically dependable to deliver profits across testing and real-time data sets.

Therefore, for the outer-shell global optimization of training sets, we have used an ASA OPTION, MULTI_MIN, which saves a user-defined number of closest local minima within a user-defined resolution of the parameters. We then examine these results under several testing sets.

5 Trading System

5.1 Use of CMI

As the CMI formalism carries the relevant information regarding the prices dynamics, we have used it as a signal generator for an automated trading system for S&P futures.

While currently we are integrating fast-response CMI signals into the trading model, next we discuss averaged CMI signals characterizing longer time scales.

Based on a previous work (Ingber 1996c) applied to daily closing data, the overall structure of the trading system consists in 2 layers, as follows: We first construct the "short-time" Lagrangian function in the Itô representation (with the notation introduced in Section 3.3)

$$L(i|i-1) = \frac{1}{2\sigma^2 F_{i-1}^{2x}} \left(\frac{dF_i}{dt} - f^F \right)^2 \tag{14}$$

with i the post-point index, corresponding to the one factor price model

$$dF = f^F dt + \sigma F^x dz(t), \tag{15}$$

where f^F and $\sigma > 0$ are taken to be constants, $F(t)$ is the S&P future price, and dz is the standard Gaussian noise with zero mean and unit standard deviation. We perform a global, maximum likelihood fit to the whole set of price data using ASA. This procedure produces the optimization parameters $\{x, f^F\}$ that are used to generate the CMI. One computational approach was to fix the diffusion multiplier σ to 1 during training for convenience, but used as free parameters in the adaptive testing and real-time fits. Another approach was to fix the scale of the volatility, using an improved model,

$$dF = f^F dt + \sigma \left(\frac{F}{< F >} \right)^x dz(t), \tag{16}$$

where σ now is calculated as the standard deviation of the price increments $\Delta F/dt^{1/2}$, and $< F >$ is just the average of the prices.

As already remarked, to enhance the CMI sensitivity and response time to local variations (across a certain window size) in the distribution of price increments, the momenta are generated applying an adaptive procedure, i.e., after each new data reading another set of $\{f^F, \sigma\}$ parameters are calculated for the last window of data, with the exponent x – a contextual indicator of the noise statistics – fixed to the value obtained from the global fit.

The CMI computed in this manner are fed into the outer shell of the trading system, where an AI-type optimization of the trading rules is executed, using ASA once again.

The trading rules are a collection of logical conditions among the CMI, prices and optimization parameters that could be window sizes, time resolutions, or trigger thresholds. Based on the relationships between CMI and optimization parameters, a trading decision is made. The cost function in the outer shell is either the overall equity or the risk-adjusted profit (essentially the return). The inner and outer shell optimizations are coupled through some of the optimization parameters (e.g., time resolution of the data, window sizes), which justifies the recursive nature of the optimization.

Next, we describe in more details the concrete implementation of this system.

5.2 Data Processing

The CMI formalism is general and by construction permits us to treat multivariate coupled markets. In certain conditions (e.g., shorter time scales of data), and also due to superior scalability across different markets, it is desirable to have a trading system for a single instrument, in our case the S&P futures contracts that are traded electronically on Chicago Mercantile Exchange (CME). The focus of our system was intra-day trading, at time scales of data used in generating the buy/sell signals from 10 to 60 secs. In particular, we here give some results obtained when using data having a time resolution Δt of 55 secs (the time between consecutive data elements is 55 secs). This particular choice of time resolution reflects the set of optimization parameters that have been applied in actual trading.

It is important to remark that a data point in our model does not necessarily mean an actual tick datum. For some trading time scales and for noise reduction purposes, data is pre-processed into sampling bins of length Δt using either a standard averaging procedure or spectral filtering (e.g., wavelets, Fourier) of the tick data. Alternatively, the data can be defined in block bins that contain disjoint sets of averaged tick data, or in overlapping bins of widths Δt that update at every $\Delta t' < \Delta t$, such that an effective resolution $\Delta t'$ shorter than the width of the sampling bin is obtained. We present here work in which we have used disjoint block bins and a standard average of the tick data with time stamps falling within the bin width.

In Figures 1 and 2 we present examples of S&P futures data sampled with 55 secs resolution. We remark that there are several time scales – from minutes to one hour – at which an automated trading system might extract profits.

Figure 1 illustrates that the profitable regions are prominent even for data representing a relatively flat market period. I.e., June 20 shows an uptrend region of about 1 hour 20 minutes and several short and long trading domains between 10 minutes and 20 minutes.

Figure 2 illustrates the sustained short trading region of 1.5 hours and several shorter long and short trading regions of about 10–20 minutes.

In both situations, there are a larger number of opportunities at time

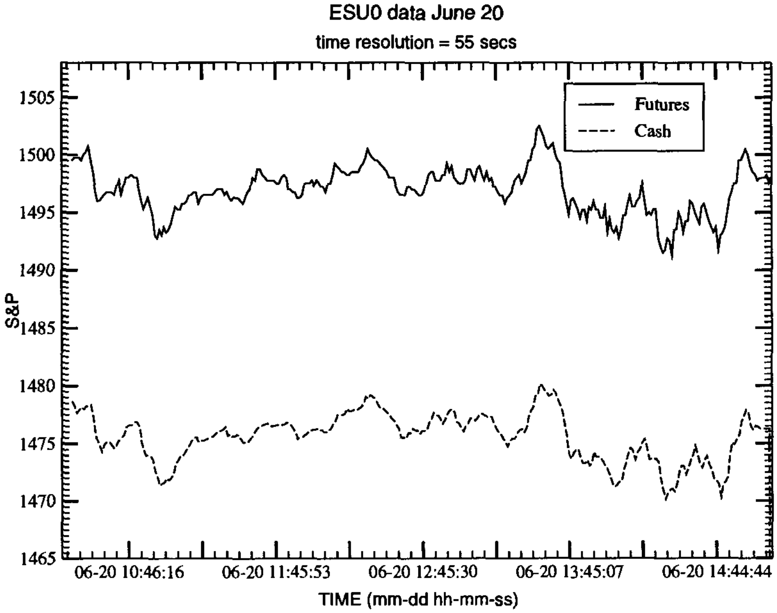


Figure 1. Futures and cash data, contract ESU0 June 20: (*solid line*) – futures; (*dashed line*) – cash.

resolutions smaller than 5 minutes.

The time scale at which we sample the data for trading is itself a parameter that is extracted from the optimization of the trading rules and of the Lagrangian cost function Eq. (14). This is one of the coupling parameters between the inner- and the outer-shell optimizations.

5.3 Inner-Shell Optimization

A cycle of optimization runs has three parts, training and testing, and finally real-time use – a variant of testing. Training consists in choosing a data set and performing the recursive optimization, which produces optimization parameters for trading. In our case there are six parameters: the time resolution Δt of price data, the length of

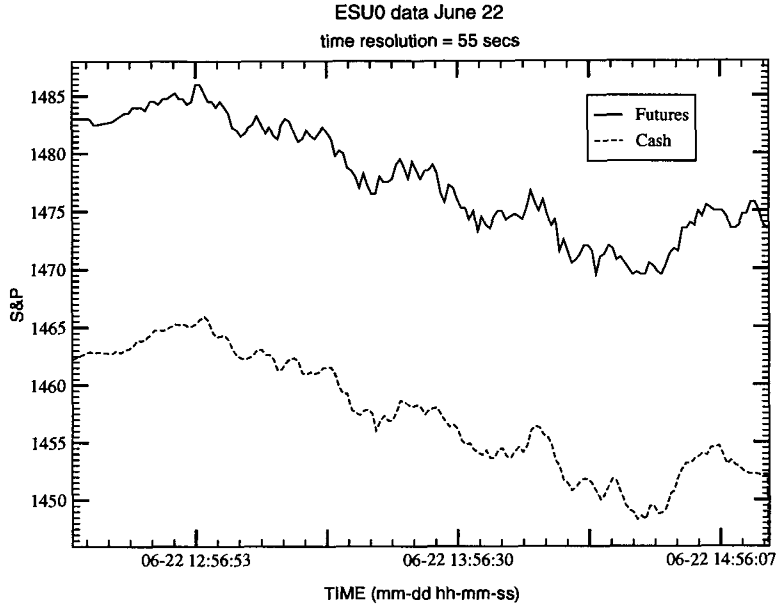


Figure 2. Futures and cash data, contract ESU0 June 22: (*solid line*) – futures; (*dashed line*) – cash.

window W used in the local fitting procedures and in computation of moving averages of trading signals, the drift f^F, volatility coefficient σ and exponent x from Eq. (15), and a multiplicative factor M necessary for the trading rules module, as discussed below.

The optimization parameters computed from the training set are applied then to various test sets and final profit/loss analyses are produced. Based on these, the best set of optimization parameters are chosen to be applied in real-time trading runs. We remark once again that a single training data set could support more than one profitable sets of parameters and can be a function of the trader's interest and the specific market dynamics targeted (e.g., short/long time scales). The optimization parameters corresponding to the global minimum in the training session may not necessarily represent the parameters that led to robust profits across real-time data.

The training optimization occurs in two inter-related stages. An inner-shell maximum likelihood optimization over all training data is performed. The cost function that is fitted to data is the effective action constructed from the Lagrangian Eq. (14) including the prefactors coming from the measure element in the expression of the short-time probability distribution Eq. (11). This is based on the fact (Langouche *et al.* 1982) that in the context of Gaussian multiplicative stochastic noise, the macroscopic transition probability $P(F, t|F', t')$ to start with the price $F'(= F_{i-1})$ at $t'(= t_{i-1})$ and reach the price $F(= F_i)$ at $t(= t_i)y$ is determined by the short-time Lagrangian Eq. (14),

$$P(F,t|F',t') = \frac{1}{(2\pi\sigma^2 F_{i-1}^{2x} dt_i)^{1/2}} \times \exp\left(-\sum_{i=1}^{N} L(i|i-1)dt_i\right), \tag{17}$$

with $dt_i = t_i - t_{i-1}$. Recall that the main assumption of our model is that price increments (or the logarithm of price ratios, depending on which variables are considered independent) could be described by a system of coupled stochastic, non-linear equations as in Eq. (9a). These equations are deceptively simple in structure, yet depending on the functional form of the drift coefficients and the multiplicative noise, they could describe a variety of interactions between financial instruments in various market conditions (e.g., constant elasticity of variance model (Cox and Ross 1976), stochastic volatility models, and so on). In particular, this type of models include the case of Black-Scholes price dynamics ($x = 1$).

In the system presented here, we have applied the model from Eq. (15). The fitted parameters were the drift coefficient f^F and the exponent x. In the case of a coupled futures and cash system, besides the corresponding values of f^F and x for the cash index, another parameter, the correlation coefficient ρ as introduced in Eq. (9a), must be considered.

5.4 Trading Rules (Outer-Shell) Recursive Optimization

In the second part of the training optimization, we calculate the CMI and execute trades as required by a selected set of trading rules based on CMI values, price data or combinations of both indicators.

Recall that three external shell optimization parameters are defined: the time resolution Δt of the data expressed as the time interval between consecutive data points, the window length W (in number of time epochs or data points) used in the adaptive calculation of CMI, and a numerical coefficient M that scales the momentum uncertainty discussed below.

At each moment a local refit of f^F and σ over data in the local window W is executed, moving the window M across the training data set and using the zeroth order optimization parameters f^F and x resulting from the inner-shell optimization as a first guess. It was found that a faster quasi-local code is sufficient for computational purposes for these adaptive updates. In more complicated models, ASA can be successfully applied recursively, although in real-time trading the response time of the system is a major factor that requires attention.

All expressions that follow can be generalized to coupled systems in the manner described in Section 3. Here we use the one factor nonlinear model given by Eq. (15). At each time epoch we calculate the following momentum related quantities:

$$\begin{aligned}
\Pi^F &= \frac{1}{\sigma^2 F^{2x}} \left(\frac{dF}{dt} - f^F \right), \\
\Pi_0^F &= -\frac{f^F}{\sigma^2 F^{2x}}, \\
\Delta \Pi^F &= \, < (\Pi^F - < \Pi^F >)^2 >^{1/2} = \frac{1}{\sigma F^x \sqrt{dt}},
\end{aligned} \tag{18}$$

where we have used $< \Pi^F >= 0$ as implied by Eqs. (15) and (14).

In the previous expressions, Π^F is the CMI, Π_0^F is the neutral line or the momentum of a zero change in prices, and $\Delta\Pi^F$ is the uncertainty of momentum. The last quantity reflects the Heisenberg principle, as derived from Eq. (15) by calculating

$$\begin{aligned} \Delta F &\equiv\, < (dF- < dF >)^2 >^{1/2} = \sigma F^x \sqrt{dt}, \\ \Delta\Pi^F \, \Delta F &\geq 1, \end{aligned} \tag{19}$$

where all expectations are in terms of the exact noise distribution, and the calculation implies the Itô approximation (equivalent to considering non-anticipative functions). Various moving averages of these momentum signals are also constructed. Other dynamical quantities, as the Hamiltonian, could be used as well. (By analogy to the energy concept, we found that the Hamiltonian carries information regarding the overall trend of the market, giving another useful measure of price volatility.)

Regarding the practical implementation of the previous relations for trading, some comments are necessary. In terms of discretization, if the CMI are calculated at epoch i, then $dF_i = F_i - F_{i-1}$, $dt_i = t_i - t_{i-1} = \Delta t$, and all prefactors are computed at moment $i - 1$ by the Itô prescription (e.g., $\sigma F^x = \sigma F_{i-1}^x$). The momentum uncertainty band $\Delta\Pi^F$ can be calculated from the discretized theoretical value Eq. (18), or by computing the estimator of the standard deviation from the actual time series of Π^F.

There are also two ways of calculating averages over CMI values: One way is to use the set of local optimization parameters $\{f^F, \sigma\}$ obtained from the local fit procedure in the current window W for all CMI data within that window (local-model average). The second way is to calculate each CMI in the current local window W with another set $\{f^F, \sigma\}$ obtained from a previous local fit window measured from the CMI data backwards W points (multiple-models averaged, as each CMI corresponds to a different model in terms of the fitting parameters $\{f^F, \sigma\}$).

The last observation is that the neutral line divides all CMI in two classes: long signals, when $\Pi^F > \Pi_0^F$, as any CMI satisfying this condition indicates a positive price change, and short signals when $\Pi^F < \Pi_0^F$, which reflects a negative price change.

After the CMI are calculated, based on their meaning as statistical momentum indicators, trades are executed following a relatively simple model: Entry in and exit from a long (short) trade points are defined as points where the value of CMIs is greater (smaller) than a certain fraction of the uncertainty band $M\,\Delta\Pi^F$ $(-M\,\Delta\Pi^F)$, where M is the multiplicative factor mentioned in the beginning of this subsection. This is a choice of a symmetric trading rule, as M is the same for long and short trading signals, which is suitable for volatile markets without a sustained trend, yet without diminishing too severely profits in a strictly bull or bear region.

Inside the momentum uncertainty band, one could define rules to stay in a previously open trade, or exit immediately, because by its nature the momentum uncertainty band implies that the probabilities of price movements in either direction (up or down) are balanced. From another perspective, this type of trading rule exploits the relaxation time of a strong market advance or decline, until a trend reversal occurs or it becomes more probable.

Other sets of trading rules are certainly possible, by utilizing not only the current values of the momenta indicators, but also their local-model or multiple-models averages. A trading rule based on the maximum distance between the current CMI data Π_i^F and the neutral line Π_0^F shows faster response to markets evolution and may be more suitable to automatic trading in certain conditions.

Stepping through the trading decisions each trading day of the training set determined the profit/loss of the training set as a single value of the outer-sell cost function. As ASA importance-sampled the outer-shell parameter space $\{\Delta t, W, M\}$, these parameters are fed into the inner shell, and a new inner-shell recursive optimization cycle begins. The final values for the optimization parameters in the training set are fixed when the largest net profit (calculated from the total equity by subtracting the transactions costs defined by the slippage factor) is realized. In practice, we have collected optimization parameters from multiple local minima that are near the global minimum (the outer-shell cost function is defined with the sign reversed) of the training set.

The values of the optimization parameters $\{\Delta t, W, M, f^F, \sigma, x\}$ resulting from a training cycle are then applied to out-of-sample test sets. During the test run, the drift coefficient f^F and the volatility coefficient σ are refitted adaptively as described previously. All other parameters are fixed. We have mentioned that the optimization parameters corresponding to the highest profit in the training set may not be the sufficiently robust across test sets. Then, for all test sets, we have tested optimization parameters related to the multiple minima (i.e., the global maximum profit, the second best profit, and so on) resulting from the training set.

We performed a bootstrap-type reversal of the training-test sets (repeating the training runs procedures using one of the test sets, including the previous training set in the new batch of test sets), followed by a selection of the best parameters across all data sets. This is necessary to increase the chances of successful trading sessions in real-time.

6 Results

6.1 Alternative Algorithms

In the previous sections we noted that there are different combinations of methods of processing data, methods of computing the CMI and various sets of trading rules that need to be tested – at least in a sampling manner – before launching trading runs in real-time:

1. Data can be preprocessed in block or overlapping bins, or forecasted data derived from the most probable transition path (Dekker 1980) could be used as in one of our most recent models.
2. Exponential smoothing, wavelets or Fourier decomposition can be applied for statistical processing. We presently favor exponential moving averages.
3. The CMI can be calculated using averaged data or directly with tick data, although the optimization parameters were fitted from preprocessed (averaged) price data.
4. The trading rules can be based on current signals (no average is performed over the signal themselves), on various averages of the CMI trading signals, on various combination of CMI data (momenta, neutral line, uncertainty band), on symmetric or asymmetric trading rules, or on mixed price-CMI trading signals.
5. Different models (one and two-factors coupled) can be applied to the same market instrument, e.g., to define complementary indicators.

The selection process evidently must consider many specific economic factors (e.g., liquidity of a given market), besides all other physical, mathematical and technical considerations. In the work presented here, as we tested our system and using previous experience, we focused toward S&P500 futures electronic trading, using block processed data, and symmetric, local-model and multiple-models trading rules based on CMI neutral line and stay-in conditions.

6.2 Trading System Design

The design of a successful electronic trading system is complex as it must incorporate several aspects of a trader's actions that sometimes are difficult to translate into computer code. Three important features that must be implemented are factoring in the transactions costs, devising money management techniques, and coping with execution deficiencies.

Generally, most trading costs can be included under the "slippage factor," although this could easily lead to poor estimates. Given that the margin of profits from exploiting market inefficiencies are thin, a high slippage factor can easily result in a non-profitable trading system. In our situation, for testing purposes we used a $35 slippage factor per buy & sell order, a value we believe is rather high for an electronic trading environment, although it represents less than three ticks of a mini-S&P futures contract. (The mini-S&P is the S&P futures contract that is traded electronically on CME.) This higher value was chosen to protect ourselves against the bid-ask spread, as our trigger price (at what price the CMI was generated) and execution price (at what price a trade signaled by a CMI was executed) were taken to be equal to the trading price. (We have changed this aspect of our algorithm in later models.) The slippage is also strongly influenced by the time resolution of the data. Although the slippage is linked to bid-ask spreads and markets volatility in various formulas (Kaufman 1998), the best estimate is obtained from experience and actual trading.

Money management was introduced in terms of a trailing stop condition that is a function of the price volatility, and a stop-loss threshold that we fixed by experiment to a multiple of the mini-S&P contract value ($200). It is tempting to tighten the trailing stop or to work with a small stop-loss value, yet we found – as otherwise expected – that higher losses occurred as the signals generated by our stochastic model were bypassed.

Regarding the execution process, we have to account for the response of the system to various execution conditions in the interaction with the electronic exchange: partial fills, rejections, uptick rule (for equity trading), and so on. Except for some special conditions, all these steps must be automated.

6.3 Some Explicit Results

Typical CMI data in Figures 3 and 4 (obtained from real-time trading after a full cycle of training-testing was performed) are related to the price data in Figures 1 and 2. We have plotted the fastest (55 secs apart) CMI values Π^F, the neutral line Π_0^F and the uncertainty band $\Delta\Pi^F$. All CMI data were produced using the optimization parameters set $\{55\text{secs}, 88\text{epochs}, 0.15\}$ of the second-best net profit obtained with a training set based on the March data of the ESM0 contract (mini-S&P June 2000 contract). We recall the meaning of the optimization parameters from 5.4: the first factor is the frequency of CMI signals (or time-step between consecutive CMIs), the second parameter is the width in time-step units of the time-window used for local statistics, and the third parameter is the scaling factor of the momentum uncertainty.

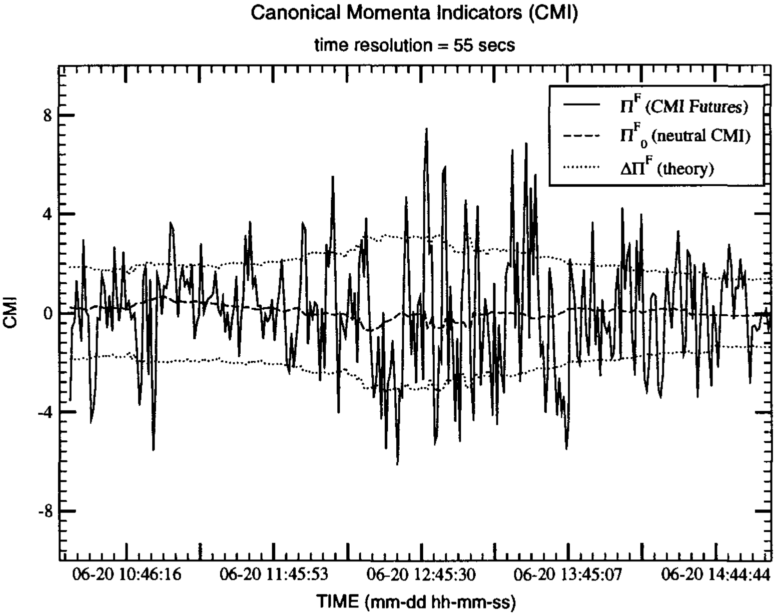


Figure 3. CMI data, real-time trading June 20: (*solid line*) – CMI; (*dashed line*) – neutral line; (*dotted line*) – uncertainty band.

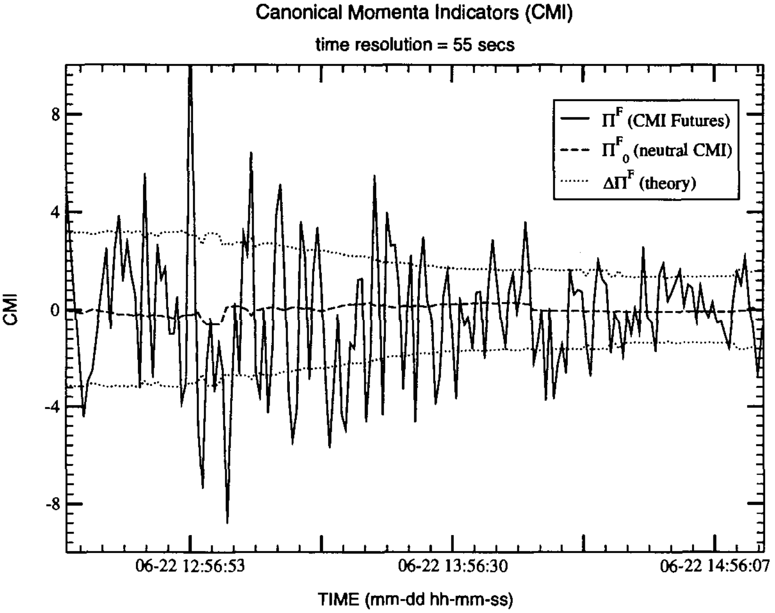


Figure 4. CMI data, real-time trading, June 22: (*solid line*) – CMI; (*dashed line*) – neutral line; (*dotted line*) – uncertainty band.

Although the CMIs exhibit an inherently ragged nature and oscillate around a zero mean value within the uncertainty band – the width of which is decreasing with increasing price volatility, as the uncertainty principle would also indicate – time scales at which the CMI average or some persistence time are not balanced about the neutral line.

These characteristics, which we try to exploit in our system, are better depicted in Figures 5 and 6.

One set of trading signals, the local-model average of the neutral line $< \Pi_0^F >$ and the uncertainty band multiplied by the optimization factor $M = 0.15$, and centered around the theoretical zero mean of the CMI, is represented versus time. Note entry points in a short trading position ($< \Pi_0^F > > \;\; M\,\Delta\Pi^F$) at around 10:41 (Figure 5

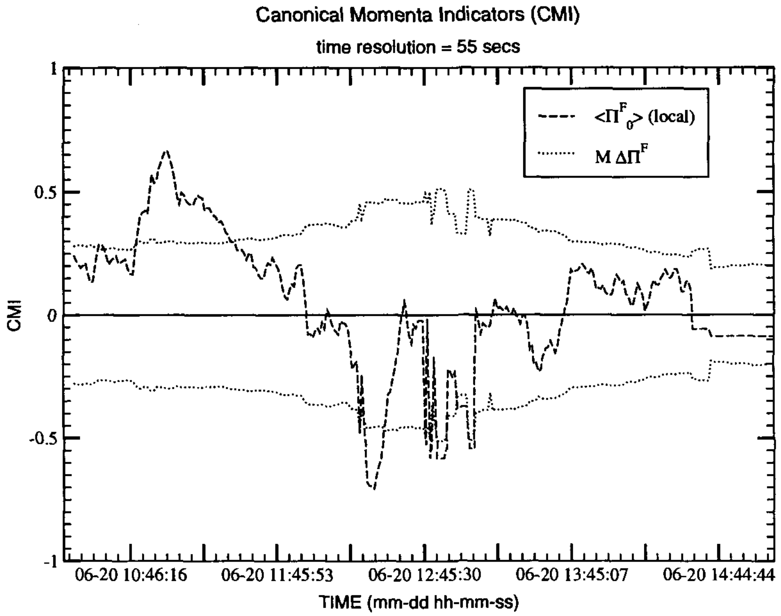


Figure 5. CMI trading signals, real-time trading June 20: (*dashed line*) – local-model average of the neutral line; (*dotted line*) – uncertainty band multiplied by the optimization parameter $M = 0.15$.

in conjunction with S&P data in Figure 1) with a possible exit at 11:21 (or later), and a first long entry ($< \Pi_0^F > < -M\,\Delta\Pi^F$) at 12:15. After 14:35, a stay long region appears ($< \Pi_0^F > < 0$), which indicates correctly the price movement in Figure 1.

In Figure 6 corresponding to June 22 price data from Figure 2, a first long signal is generated at around 12:56 and a first short signal is generated at 14:16 that reflects the long downtrend region in Figure 2. Due to the averaging process, a time lag is introduced, reflected by the long signal at 12:56 in Figure 4, related to a past upward trend seen in Figure 2; yet the neutral line relaxes rather rapidly (given the 55-second time resolution and the window of $88 \approx 1.5$ hour) toward the uncertainty band. A judicious choice of trading rules, or avoiding standard averaging methods, helps in controlling this lag problem.

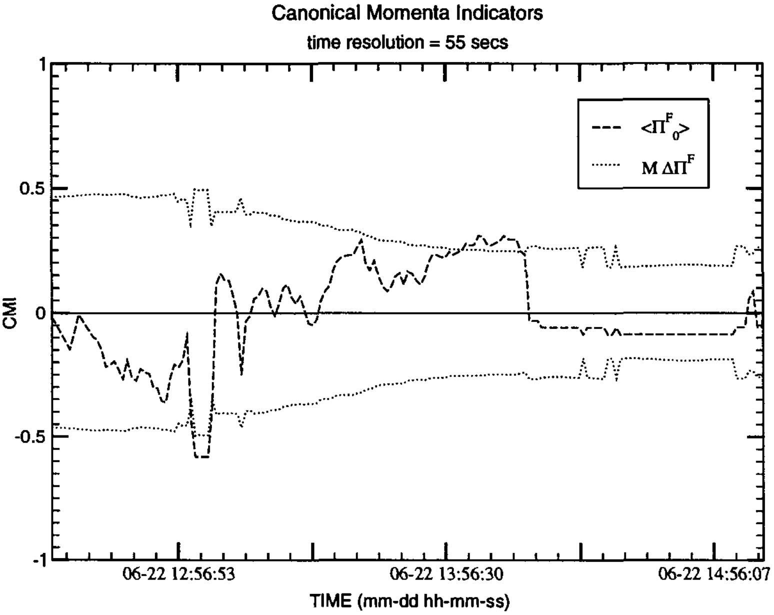


Figure 6. CMI trading signals, real-time trading June 22: (*dashed line*) – local-model average of the neutral line; (*dotted line*) – uncertainty band multiplied by the optimization parameter $M = 0.15$.

Recall that the trading rules presented are symmetric (the long and short entry/exit signals are controlled by the same M factor), and we apply a stay-long condition if the neutral-line is below the average momentum $< \Pi^F >= 0$ and stay-short if $< \Pi_0^F >> 0$. The drift f^F and volatility coefficient σ are refitted adaptively and the exponent x is fixed to the value obtained in the training set. Typical values are $f^F \in \pm[0.003 : 0.05]$, $x \in \pm[0.01 : 0.03]$. During the local fit, due to the shorter time scale involved, the drift may increase by a factor of ten, and $\sigma \in [0.01 : 1.2]$.

We note that the most robust optimization factors – in terms of maximum cumulative profit resulted for all test sets – do not correspond to the maximum profit in the training sets: For the local-model rules, the optimum parameters are $\{55, 88, 0.15\}$, and for the multiple models rules the optimum set is $\{45, 72, 0.2\}$, both realized by a four-days training set from the March 2000 mini-S&P contract (Ingber and Mondescu 2001).

Other observations are that, for the data presented here, the multiple-models averages trading rules consistently performed better and are more robust than the local-model averages trading rules. The number of trades is similar, varying between 15 and 35 (eliminating cumulative values smaller than 10 trades), and the time scale of the local fit is rather long in the 30 minutes to 1.5 hour range. In the current set-up, this extended time scale implies that is advisable to deploy this system as a trader-assisted tool.

An important factor is the average length of the trades. For the type of rules presented in this work, this length is of several minutes, up to one hour, as the time scale of the local fit window mentioned above suggested.

Related to the length of a trade is the length of a winning long/short trade in comparison to a losing long/short trade. Our experience indicates that a ratio of 2:1 between the length of a winning trade and the length of a losing trade is desirable for a reliable trading system. Here, using the local-model trading rules seems to offer an advantage, although this is not as clear as one would expect. More details regarding the data and results obtained with the trading system are given in our earlier work (Ingber and Mondescu 2001).

7 Conclusions

7.1 Main Features

The main stages of building and testing this system are summarized in the followiong lines:

1. We developed a multivariate, nonlinear statistical mechanics

model of S&P futures and cash markets, based on a system of coupled stochastic differential equations.

2. We constructed a two-stage, recursive optimization procedure using methods of ASA global optimization: An inner-shell extracts the characteristics of the stochastic price distribution and an outer-shell generates the technical indicators and optimize the trading rules.
3. We trained the system on different sets of data and retained the multiple minima generated (corresponding to the global maximum net profit realized and the neighboring profit maxima).
4. We tested the system on out-of-sample data sets, searching for most robust optimization parameters to be used in real-time trading. Robustness was estimated by the cumulative profit/loss across diverse test sets, and by testing the system against a bootstrap-type reversal of training-testing sets in the optimization cycle.

   Modeling the market as a dynamical physical system makes possible a direct representation of empirical notions as market momentum in terms of CMI derived naturally from our theoretical model. We have shown that other physical concepts as the uncertainty principle may lead to quantitative signals (the momentum uncertainty band $\Delta\Pi^F$) that captures other aspects of market dynamics and which can be used in real-time trading.
5. We presented and discussed the main aspects of developing an internet-based interface (API) for connecting a proprietary trading system to an exchange.

7.2 Summary

We have presented an internet-enabled trading system with its two components: the connection API and the computational trading engine.

The trading engine is composed of an outer-shell trading-rule model and an inner-shell nonlinear stochastic dynamic model of the market of interest, S&P500. The inner-shell is developed adhering to the mathematical physics of multivariate nonlinear statistical mechanics, from which we develop indicators for the trading-rule model, i.e., canonical momenta indicators (CMI). We have found that keeping our model faithful to the underlying mathematical physics is not a limiting constraint on profitability of our system; quite the contrary.

An important result of our work is that the ideas for our algorithms, and the proper use of the mathematical physics faithful to these algorithms, must be supplemented by many practical considerations en route to developing a profitable trading system. For example, since there is a subset of parameters, e.g., time resolution parameters, shared by the inner- and outer-shell models, recursive optimization is used to get the best fits to data, as well as developing multiple minima with approximate similar profitability. The multiple minima often have additional features requiring consideration for real-time trading, e.g., more trades per day increasing robustness of the system, and so on. The nonlinear stochastic nature of our data required a robust global optimization algorithm. The output of these parameters from these training sets were then applied to testing sets on out-of-sample data. The best models and parameters were then used in real-time by traders, further testing the models as a precursor to eventual deployment in automated electronic trading.

We have used methods of statistical mechanics to develop our inner-shell model of market dynamics and a heuristic AI type model for our outer-shell trading-rule model, but there are many other candidate (quasi-)global algorithms for developing a cost function that can be used to fit parameters to data, e.g., neural nets, fractal scaling models, and so on. To perform our fits to data, we selected an algorithm, Adaptive Simulated Annealing (ASA), that we were familiar with, but there are several other candidate algorithms that likely would suffice, e.g., genetic algorithms, tabu search, and so on.

We have shown that a minimal set of trading signals (the CMI, the neutral line representing the momentum of the trend of a given time window of data, and the momentum uncertainty band) can generate a rich and robust set of trading rules that identify profitable domains of trading at various time scales. This is a confirmation of the hypothesis that markets are not efficient, as noted in other studies (Brock *et al.*1992, Ingber 1984, 1996c).

7.3 Future Directions

Although this chapter focused on trading of a single instrument, the futures S&P 500, the code we have developed can accommodate trading on multiple markets. For example, in the case of tick-resolution coupled cash and futures markets, which was previously prototyped for inter-day trading (Ingber 1996b,c),the utility of CMI stems from three directions:

1. The inner-shell fitting process requires a global optimization of all parameters in both futures and cash markets.
2. The CMI for futures contain, by our Lagrangian construction, the coupling with the cash market through the off-diagonal correlation terms of the metric tensor. The correlation between the futures and cash markets is explicitly present in all futures variables.
3. The CMI of both markets can be used as complimentary technical indicators for trading in futures market.

Several near term future directions are of interest:

- finalizing the production-level order execution API,
- orienting the system toward shorter trading time scales (10-30 secs) more suitable for electronic trading,
- introducing fast response "averaging" methods and time scale identifiers (exponential smoothing, wavelets decomposition),
- identifying mini-crashes points using renormalization group techniques,
- investigating the use of CMI in pattern-recognition based trading rules,
- exploring the use of forecasted data evaluated from most probable transition path formalism.

7.4 Standard Disclaimer

We must emphasize that there are no claims that all results are positive or that the present system is a safe source of riskless profits. There as many negative results as positive, and a lot of work is necessary to extract meaningful information.

Acknowledgments

We thank Donald Wilson for his financial support. We thank K.S. Balasubramaniam and Colleen Chen for their programming support and participation in formulating parts of our trading system. Data was extracted from the DRW Reuters feed.

References

Archipelago (2001), http://www.tradearca.com .

Azoff, E. (1994), *Neural Network Time Series Forecasting of Financial Markets,* Wiley & Sons, New York, NY.

Bachelier, L. (1900), "Théorie de la spéculation," *Annales de l'Ecole Normale Supérieure,* vol. 17, pp. 21-86.

Barabino, G.P., Barabino, G.S., Bianco, B., and Marchesi, M. (1980), "A study on the performances of simplex methods for function minimization," *Proc. IEEE Int. Conf. Circuits and Computers,* pp. 1150-1153.

Brock, W., Lakonishok, J., and LeBaron, B. (1992), " Simple technical trading rules and the stochastic properties of stock returns," *J. Finance,* vol. 47, no. 5, pp. 1731-1763.

Brown, P., Kleidon, A.W., and Marsh, T.A. (1983), "New evidence on the nature of size-related anomalies in stock prices," *J. Fin. Econ.*, vol. 12, pp. 33-56.

Burghardt, G. (2001), "Whassup?," *Futures Industry Magazine*, February/March.

Cheng, K.S. (1972), "Quantization of a general dynamical system by Feynman's path integration formulation," *J. Math. Phys.*, vol. 13, pp. 1723-1726.

CME Market Data API (2001), http://www.cme.com/electronic/mdapi .

Cohen, B. (1994), "Training synaptic delays in a recurrent neural network," M.Sc. Thesis, Tel-Aviv University, Tel-Aviv, Israel (unpublished).

Cox, J.C. and Ross, S.A. (1976), "The valuation of options for alternative stochastic processes," *J. Fin. Econ.*, vol. 3, pp. 145-166.

Cozzio-Buëler, R.A. (1995), *The design of neural networks using a prior knowledge*, Ph.D. dissertation, Swiss Fed. Inst. Technol., Zurich, Switzerland.

Dekker, H. (1979), "Functional integration and the Onsager-Machlup Lagrangian for continuous Markov processes in Riemannian geometries," *Phys. Rev. A*, vol. 19, pp. 2102-2111.

Dekker, H. (1980), "On the most probable transition path of a general diffusion process," *Phys. Lett. A*, vol. 80, pp. 99-101.

eSignal (Interactive Data Corporation) (2001), http://www.esignal .com .

Eurex Values API (2001), http://www.eurexchange.com/marketplace.

FIX Protocol Organization (2000), http://www.fixprotocol.org .

Gamma, E., Helm, R:, Johnson, R., and Vlissides, J. (1994), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.

Gardiner, C. (1983), *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, Springer-Verlag, Berlin, Germany.

Gately, E. (1996), *Neural Networks for Financial Forecasting*, Wiley & Sons, New York, NY.

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images," *IEEE Trans. Patt. Anal. Mac. Int.*, vol. 6, no. 6, pp. 721-741.

Graham, R. (1978), "Path-integral methods in nonequilibrium thermodynamics and statistics," in Garrido, L., Seglar, P., and Shepherd, P.J. (eds.), *Stochastic Processes in Nonequilibrium Systems*, Springer, New York, NY, pp. 82-138.

Haken, H. (1983), *Synergetics*, 3rd ed., Springer, New York.

Harrison, J.M. and Kreps, D. (1979), "Martingales and arbitrage in multiperiod securities markets," *J. Econ. Theory*, vol. 20, pp. 381-408.

Hull, J.C. (2000), *Options, Futures, and Other Derivatives*, 4th ed., Prentice Hall, Upper Saddle River, NJ.

Ilinsky, K. and Kalinin, G. (1997), "Black-Scholes equation from gauge theory of arbitrage," Technical Report, LANL (unpublished), http://xxx.lanl.gov/hep-th/9712034 .

Indiveri, G., Nateri, G., Raffo, L., and Caviglia, D. (1993), "A neural network architecture for defect detection through magnetic inspection," Report, University of Genova, Genova, Italy (unpublished).

Ingber, L. (1984), "Statistical mechanics of nonlinear nonequilibrium financial markets," *Math. Modelling*, vol. 5, no. 6, pp. 343-361, http://www.ingber.com/markets84_statmech.ps.gz .

Ingber, L. (1985), "Statistical mechanics of neocortical interactions: stability and duration of the 7+-2 rule of short-term-memory capacity," *Phys. Rev. A*, vol. 31, pp. 1183-1186, http://www.ingber.com/smni85_stm.ps.gz .

Ingber, L. (1989), "Very fast simulated re-annealing," *Math. Comput. Modelling*, vol. 12, no. 8, pp. 967-973, http://www.ingber.com /asa89_vfsr.ps.gz .

Ingber, L. (1991), "Statistical mechanics of neocortical interactions: a scaling paradigm applied to electroencephalography," *Phys. Rev. A*, vol. 44, no. 6, pp. 4017-4060, http://www.ingber.com /smni91_eeg.ps.gz .

Ingber, L. (1993a), "Adaptive Simulated Annealing (ASA)," Tech. Report "Global optimization C-code," Caltech Alumni Association, Pasadena, CA, http://www.ingber.com/#ASA-CODE .

Ingber, L. (1993b), "Simulated annealing: practice versus theory," *Math. Comput. Modelling*, vol. 18, no. 11, pp. 29-57, http://www.ingber.com/asa93_sapvt.ps.gz .

Ingber, L. (1996a), "Adaptive simulated annealing (ASA): lessons learned," *Control and Cybernetics*, vol. 25, no. 1, pp. 33-54, invited paper on "Simulated Annealing Applied to Combinatorial Optimization," http://www.ingber.com/asa96_lessons.ps.gz .

Ingber, L. (1996b), "Canonical momenta indicators of financial markets and neocortical EEG," in Amari, S.I., Xu, L., King, I., and Leung, K.-S. (eds.), *Progress in Neural Information Processing*, Springer, New York, pp. 777-784, http://www.ingber.com /markets96_ momenta.ps.gz .

Ingber, L. (1996c), "Statistical mechanics of nonlinear nonequilibrium financial markets: applications to optimized trading," *Math. Computer Modelling*, vol. 23, no. 7, pp. 101-121, http://www.ingber.com/markets96_trading.ps.gz .

Ingber, L. (2000), "High-resolution path-integral development of financial options," *Physica A*, vol. 283, no. 3/4, pp. 529-558, http://www.ingber.com/markets00_highres.ps.gz .

Ingber, L. and Mondescu, R.P. (2001), "Optimization of trading physics models of markets," *IEEE Trans. on Neural Networks*, vol. 12, no. 4, pp. 776-790, http://www.ingber.com/markets01_optim_trading.pdf .

Ingber, L. and Rosen, B. (1993), "Genetic algorithms and very fast simulated reannealing: a comparison," *Oper. Res. Management Sci.*, vol. 33, no. 5, p. 523.

Ingber, L., Wehner, M.F., Jabbour, G.M., and Barnhill, T.M. (1991), "Application of statistical mechanics methodology to term-structure bond-pricing models," *Math. Comput. Modelling*, vol. 15, no. 11, pp. 77-98, http://www.ingber.com /markets91_interest.ps.gz .

Jensen, M.C. (1978), "Some anomalous evidence regarding market efficiency, an editorial introduction," *J. Finan. Econ.*, vol. 6, pp. 95-101.

Johansen, A., Sornette, D., and Ledoit, O. (1999), "Predicting financial crashes using discrete scale invariance," *J. Risk*, vol. 1, no. 4, pp. 5-32.

Kaufman, P.J. (1998), *Trading Systems and Methods*, 3rd ed., John Wiley & Sons, New York, NY.

Kirkpatrick, S., Gelatt Jr., C.D., and Vecchi, M. (1983), "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680.

Laloux, L., Cizeau, P., Bouchaud, J.-P., and Potters, M. (1999), "Noise dressing of financial correlation matrices," *Phys. Rev. Lett.*, vol. 83, pp. 1467-1470.

Langouche, F., Roekaerts, D., and Tirapegui, E. (1979), "Discretization problems of functional integrals in phase space," *Phys. Rev. D*, vol. 20, pp. 419-432.

Langouche, F., Roekaerts, D., and Tirapegui, E. (1980), "Short derivation of Feynman Lagrangian for general diffusion process," *J. Phys. A*, vol. 113, pp. 449-452.

Langouche, F., Roekaerts, D., and Tirapegui, E. (1982), *Functional Integration and Semiclassical Expansions*, Reidel, Dordrecht, The Netherlands.

Mandelbrot, B.B. (1971), "When can price be arbitraged efficiently? A limit to the validity of the random walk and martingale models," *Rev. Econ. Statist.*, vol. 53, pp. 225-236.

Mandelbrot, B.B. (1997), *Fractals and Scaling in Finance*, Springer-Verlag, New York, NY.

Mantegna, R.N. and Stanley, H.E. (1996), "Turbulence and financial markets," *Nature*, vol. 383, pp. 587-588.

Mayer, D.G., Pepper, P.M., Belward, J.A., Burrage, K. and Swain, A.J. (1996), "Simulated annealing – a robust optimization technique for fitting nonlinear regression models," *Proceedings "Modelling, Simulation and Optimization" Conference*, International Association of Science and Technology for Development (IASTED), 6-9 May, Gold Coast.

Merton, R.C. (1973), "An intertemporal capital asset pricing model," *Econometrica*, vol. 41, pp. 867-887.

Nelder, J.A. and Mead, R. (1964), "A simplex method for function minimization," *Computer J. (UK)*, vol. 7, pp. 308-313.

Oksendal, B. (1998), *Stochastic Differential Equations*, Springer, New York, NY.

Peters, E. (1991), *Chaos and Order in the Capital Markets*, Wiley & Sons, New York, NY.

Pliska, S.R. (1997), *Introduction to Mathematical Finance*, Blackwell, Oxford, UK.

Rogue Wave Software (2001), http://www.roguewave.com .

Sakata, S. and White, H. (1998), "High breakdown point conditional dispersion estimation with application to S&P 500 daily returns volatility," *Econometrica*, vol. 66, pp. 529-567.

Szu, H. and Hartley, R. (1987), "Fast simulated annealing," *Phys. Lett. A*, vol. 122, no. 3/4, pp. 157-162.

Taylor, S.J. (1982), "Tests of the random walk hypothesis against a price-trend hypothesis," *J. Finan. Quant. Anal.*, vol. 17, pp. 37-61.

Wofsey, M. (1993), "Technology: shortcut tests validity of complicated formulas," *The Wall Street Journal*, vol. 222, no. 60, p. B1.

Chapter 10

Implementing and Maintaining a Web Case-Based Reasoning System for Heating Ventilation and Air Conditioning Systems Sales Support

I. Watson

This chapter describes the implementation and maintenance of a Case-Based Reasoning system to support HVAC sales staff operating in remote locations. The system operates on the world wide web and uses XML as a communications protocol between client and server side Java applets. The chapter describes the motivation for the system, its implementation, trial, roll-out detailing the benefits it has provided to the company. The chapter then details how the system's case-base grew rapidly causing a problem of case redundancy. A simple algorithm to identify and remove redundant cases is described along with the results of applying it to the case-base. Case obsolescence was also encountered and partially remedied using DBMS techniques. The chapter analyses the case-base maintenance required by the system in terms of Richter's knowledge containers and Leake and Wilson's CBM framework and contrasts this case study with experience from NEC and Daimler-Chrysler. The chapter observes that had maintenance of the case-base been considered more explicitly during system design and implementation, some of the resulting maintenance would have been unnecessary.

1 Introduction

Western Air is a distributor of HVAC (heating ventilation and air conditioning systems in Australia with a turnover in 1997 of $40 million (Australian dollars AUD). Based in Fremantle the company operates mainly in Western Australia, including isolated communities in the Great Sandy, Great Victoria, and Gibson deserts; a geographic area of nearly two million square miles. The systems supported range from simple residential HVAC systems to complex installations in new build and existing factories and office buildings.

2 The Problem

Western Air has a distributed sales force numbering about 100. The majority of staff do not operate from head office but are independent, working from home or a mobile base (typically their car). In fact many sales staff seldom visit Fremantle. The sales staff are technically trained being required to take a four week training course covering most aspects of the systems they supply. They do not install systems, this work is done by specialist sub-contractors.

Simple installations, such as a set of window or exterior wall mounted AC box units can be easily specified, and priced by even the most novice sales staff. However, the specification and cost estimation of more complex systems involving roof mounted AC units, ducting, fans and sensors requires the expertise of a fully qualified HVAC engineer. Western Air employs about five fully qualified engineers (two of whom are the firms owners). Until recently, sales staff in the field would gather the prospective customer's requirements using standard form and proprietary software, take measurements of the property and fax the information to Western Air in Fremantle. A qualified engineer would then specify the HVAC system. Typically the engineer would have to phone the sales staff and ask for additional information. Usually the sales staff would have to make several visits to the customer's building and pass additional information back to the engineer.

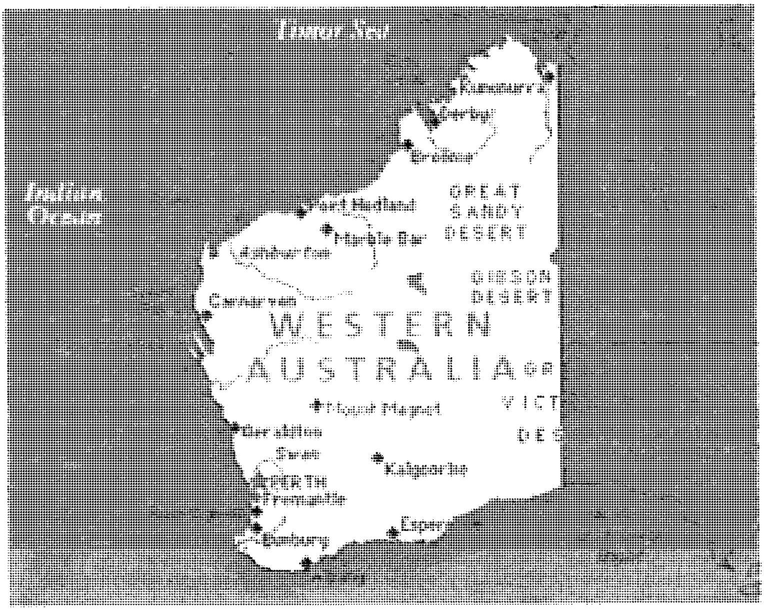


Figure 1. Map of Western Australia.

The engineer would then specify and cost the installation and a quote would be prepared and faxed to the sales staff. The sales staff would forward the quote to the customer and is empowered to negotiate on price within set margins. However, if the customer then decided that perhaps they needed fewer sensors or now only wanted certain zones in the building cooled the sales staff would have to contact the engineer and the cycle would repeat.

This process could take several weeks if the engineers were busy with other work and during this process the sales staff may be detained "beyond the Black Stump" (Australian slang for "a remote place" such as Kununurra in the far north) or lose the sale to a competitor.

Engineers when preparing specifications and quotes use a variety of specialized software to calculate HVAC loadings and made extensive use of previous installations. In particular Western Air felt that basing a quote on the price of a previous similar installation gave a more accurate estimation than using prices based on proprietary software, catalogue equipment prices and standard labor rates. However, they were aware that they were not making use of all their past work. They had nearly ten thousand system installation files but most engineers only made use of their favorite few dozen. To try to help engineers make use of all the past installations a database was created to let engineers search for past installations. The database records contained about 30 to 60 fields describing the key features of each installation and then a list of file names for the full specification. These might be MS Word documents, Excel files or AutoCAD files.

Initially the engineers liked the database and it increased the number of past installations they used as references. However, after the honeymoon ended, they started to complain that it was too hard to query across more than two or three fields at once. And that querying across ten or more fields was virtually impossible. In fact most of them admitted to using the database to laboriously browse through past installations until they found one that looked similar to their requirements.

3 The Solution

Western Air realized they wanted a system that could find similar installations without making the query too complex for the engineers. By chance they employed a new engineer who had done a computing Masters degree in the UK that had introduced CBR to him. Web based CBR applications have been demonstrated for a few years now such as the FAQFinder and FindME systems (Hammond *et al.* 1996) and those at Broderbund and Lucas Arts

(Watson 1997). They therefore felt that CBR on the web was suited for this project and contacted AI-CBR for advice.

Western Air decided that merely improving the efficiency of the engineers in Fremantle would not solve the whole problem. Ideally they would like the sales staff to be able to give fast accurate estimates to prospective customers on the spot. However, they were aware that there was a danger that the less knowledgeable sales staff might give technically incorrect quotes.

The solution they envisaged was to set up a web site that sales staff could access from anywhere in the country. Through a forms interface the prospect's requirements could be input and would be passed to a CBR system that would search the library of past installations and retrieve similar installations. Details of the similar installations along with the ftp addresses of associated files would then be available to the sales staff by ftp. The sales staff could then download the files and use these to prepare an initial quote. All this information would then be automatically passed back to an engineer to authorize or change if necessary. Once an installation was completed its details would be added to the library and its associated files placed on the ftp server.

3.1 Expected Benefits

Western Air expected the following benefits:

- A reduction in the time taken to turn around sales quotes from an average of five days to two days. It was estimated this might save approximately $250,000 a year.
- An increase in the accuracy of their estimates allowing them to judge their margins better and be more competitive. If they were able to reliably increase their margins (whilst keeping their quotes competitive) by 1% it would increase Western Air's profits by $500,000 a year.

3.2 The Team

The development team comprised:

- a senior engineer from Western Air (one of the firms owners) as project champion,
- an engineer from Western Air to act as project manager and domain expert,
- a consultant Java/HTML programmer,
- a consultant from AI-CBR to advise on CBR issues (resident in the UK), and
- a part-time data entry clerk.

3.3 Implementation Plan

The project had the direct involvement of one of the firms owners so management commitment was not a problem. It was also decided that creating a partially functional prototype was not sensible since the system would either work or not. However, a carefully controlled and monitored trial was considered essential for two reasons:

1. It was still not certain that sales staff could create technically sound first estimates and therefore a small carefully monitored trial was essential to avoid losing the firm money.
2. There were resource implications since although all sales staff had portable PCs, some were old 486 Windows 3.1 machines and few had modems or Internet accounts.

A fixed (non-negotiable) budget was given to the project of $50,000 and it was decided that six months would be given for development and trial of the system. The project started in October of 1997 and the trial was planned for March of 1998.

It was decided initially to deal with moderately complex residential HVAC systems because it was felt that this would provide a reasonable test of the system without undue risk. Western Air felt that it was commercially unwise to risk experimentation on high value commercial contracts.

3.4 Hardware & Software

A Windows NT server was purchased to act as both web and ftp server. It was decided to keep the HVAC information in the original database (MS Access) since this would remove the need to create a new case-library. An evaluation of commercially available CBR tools with web facilities was undertaken including Inference's CasePoint WebServer, ServiceSoft's WebAdvisor, and Brightware's Art*Enterprise (Watson 1997). Inference and ServiceSoft's products were eliminated because they are designed for diagnostic customer support and predominately handle textual case data. Brightware's product although technically suitable was rejected on cost grounds.

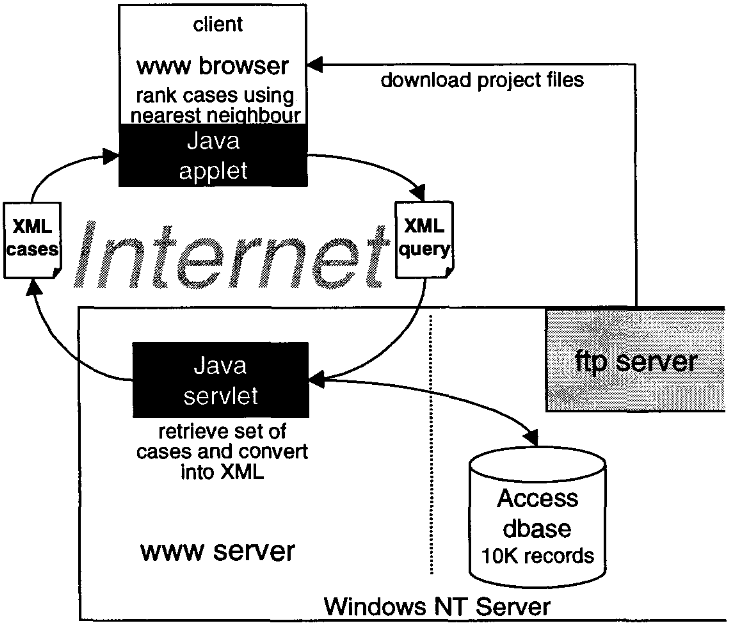


Figure 2. System architecture.

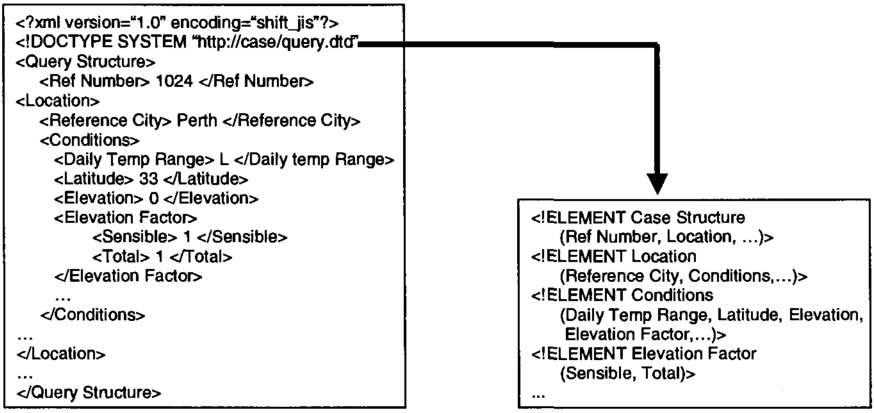
Since a simple nearest neighbor retrieval algorithm would almost certainly suffice implementing our own system was a viable option. Java (Visual Café) was chosen as the implementation language for both the client and server side elements of the CBR system. XML (eXtensible Markup Language) (W3 Consortium 1997) was used as the communication language between client and server-side applets. The World-Wide Web Consortium (W3C) finalized XML 1.0 in December 1997 as a potential successor to HTML. HTML provides a fixed and limited tag set, whereas XML authors can define an unlimited number of tags. XML therefore can incorporate commands that can be interpreted by applications and user-defined attribute:value pairs. Thus, XML is a natural communications standard for distributed intelligent systems operating on the web. It is relatively simple to develop Java applets that can interpret XML and display the results in the browser window.

3.5 System Architecture

On the sales staff (client) side a Java applet is used to gather the customer's requirements and send them as XML to the server. On the server side another Java applet (a servlet) uses this information to query the Access database to retrieve a set of relevant records. The Java servlet then converts these into XML and sends them to the client side applet that uses a nearest neighbor algorithm to rank the set of cases.

3.6 Case Representation

Cases are stored permanently within the Access database as conventional database records. Each record (case) comprises between 30 to 60 fields used for retrieval and many more used to describe the HVAC installations. In addition, links to other files on the ftp server are included to provide more detailed descriptions.



```
<?xml version="1.0" encoding="shift_jis"?>
<!DOCTYPE SYSTEM "http://case/query.dtd"
<Query Structure>
   <Ref Number> 1024 </Ref Number>
<Location>
   <Reference City> Perth </Reference City>
   <Conditions>
     <Daily Temp Range> L </Daily temp Range>
     <Latitude> 33 </Latitude>
     <Elevation> 0 </Elevation>
     <Elevation Factor>
          <Sensible> 1 </Sensible>
          <Total> 1 </Total>
     </Elevation Factor>
     ...
   </Conditions>
...
</Location>
...
</Query Structure>
```

query.xml

```
<!ELEMENT Case Structure
    (Ref Number, Location, ...)>
<!ELEMENT Location
    (Reference City, Conditions,...)>
<!ELEMENT Conditions
    (Daily Temp Range, Latitude, Elevation,
     Elevation Factor,...)>
<!ELEMENT Elevation Factor
    (Sensible, Total)>
...
```

query.dtd

Figure 3. A sample of the XML case description.

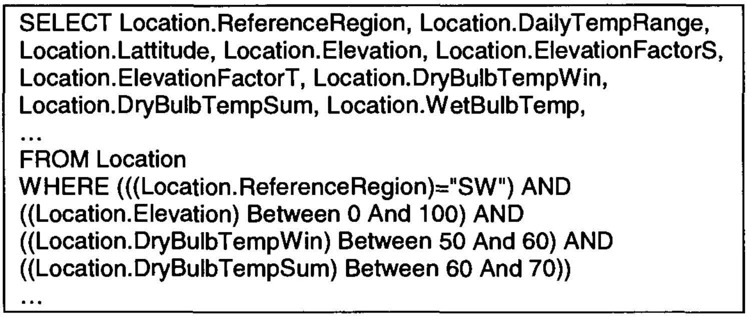
Once retrieved from the database the records are ranked by a nearest neighbor algorithm and dynamically converted into XML for presentation to the client browser. A similar XML case representation to that used by Shimazu (1998) is used by our system. XML pages can contain any number of user defined tags defined in a document type definition (DTD) file. Tags are nested hierarchically from a single root tag that can contain any number of child tags. Any child tag in turn can contain any number of child tags. Each tag contains a begin statement (e.g. <Case>) and an end statement (e.g. </Case>). This is illustrated in Figure 3.

3.7 Case Acquisition

Western Air had already put a considerable amount of effort into developing their HVAC installation database, which was used as the case library for our system. Consequently the project was fortunate in not having to acquire cases or pre-process them. However, knowledge engineering was required to create similarity metrics and obtain default weightings for the retrieval algorithm. This was not surprising as the similarity measure is one of the most important knowledge containers of any CBR system (Richter 1998).

3.8 Case Retrieval

Case retrieval is a two stage process. In stage one the customer's requirements are relaxed through a process of query relaxation. What this process does is to take the original query and relax certain terms in it to ensure that a useful number of records are retrieved from the database. This is similar to the technique used by Kitano and Shimazu (1996) in the SQUAD system at NEC.



```
SELECT Location.ReferenceRegion, Location.DailyTempRange,
Location.Lattitude, Location.Elevation, Location.ElevationFactorS,
Location.ElevationFactorT, Location.DryBulbTempWin,
Location.DryBulbTempSum, Location.WetBulbTemp,
...
FROM Location
WHERE (((Location.ReferenceRegion)="SW") AND
((Location.Elevation) Between 0 And 100) AND
((Location.DryBulbTempWin) Between 50 And 60) AND
((Location.DryBulbTempSum) Between 60 And 70))
...
```

Figure 4. Example of an SQL query that has been relaxed.

For example, let us assume that we are trying to retrieve details of properties in or near Perth in the South West of the state. An SQL query that just used "Perth" as a search term might be too restrictive. Using a symbol hierarchy our system knows that Perth is in the South West of the state so the query is relaxed to "Where (((Location,ReferenceRegion) = "SW")...)). This query will include installations from Perth, Fremantle, Rockingham and surroundings. Similarly specific elevations or temperatures can be relaxed to ranges (e.g. "Between 60 And 70").

Determining exactly how the query could be relaxed involved knowledge engineering and for example involved creating symbol hierarchies for location, building types and usage. The Java servlet queries the database to retrieve a set of broadly similar records. If enough records are not retrieved (five is considered to be enough) the query is relaxed further. If too many records are retrieved (too many is more than 20) then the query is made firmer to reduce the number. Once a sufficient set of records has been retrieved they are converted into XML and sent to the client-side applet.

In the second stage the small set of retrieved records are compared by the client-side applet with the original query and similarity is calculated using a simple nearest neighbor algorithm:

$$Similarity(T,S) = \sum_{i=1}^{n} f(T_i, S_i) \times w_i\,, \tag{1}$$

where:

T is the target case,
S is the source case,
n is the number of features in each case,
i is an individual feature from 1 to *n*,
f is a similarity function for feature *i* in cases *T* and *S*, and
w is the importance weighting of feature *i*.

Western Air expressed some surprise at the necessity for this second step and did not see the need for calculating a similarity score. Initially they felt that it would be sufficient to just show the small set of retrieved records. However, during the trials the sales staff found that the similarity score was useful. Moreover, once they understood the principle they could override the default feature weightings if they wished which they also found useful. Changing the weightings let them reflect either the customer's preferences or their own experience.

3.9 Case Retention

Once an HVAC installation is completed its details are added to the Access database and its associated files placed on the ftp server. Having a database management system for the case repository has proved very helpful since it makes it easier to generate management reports and ensure data integrity. It would be almost impossible to maintain a collection of 10,000 cases without a DBMS.

3.10 Interface Design

The interface to the system is a standard Java enabled web browser (Netscape or Internet Explorer). The forms within the Java applet were designed to look as similar to the original forms, HVAC specification tools and reports that the sales staff were already familiar with. Microsoft FrontPage was the primary tool used to create the web site.

3.11 Testing

Two weeks before trial five test scenarios were created by the project's champion. These were representative of the range of more complex residential installations the system would be expected to handle in use. The project's champion an experienced HVAC engineer knew what the correct answers should be. These were given to the five sales staff who would initially use the system and they were asked to test the system. Out of the 25 tests (5x5) 22 were correct. Although the remaining three were not specified as expected they were felt to be technically acceptable solutions.

3.12 Roll-out

The system was rolled out for trial to the five sales staff in March of 1998. At first the project's champion monitored all the projects that were being processed by the system. As his confidence grew in the system this was reduced to a weekly review.

Acceptance of the system from the five sales staff was very good once they understood what it was doing. At first they expected it to be calculating HVAC loads as the software they had previously used had done. Once they understood that it was interrogating

Western Air's database of HVAC installations they understood how it could be used to provide them with much more than just HVAC loads. During the month's trial the system dealt with 63 installations all of which were felt to be technically sound. The sales staff had not had to use the expertise of the HVAC engineers at all for this work although the engineers checked the final specifications.

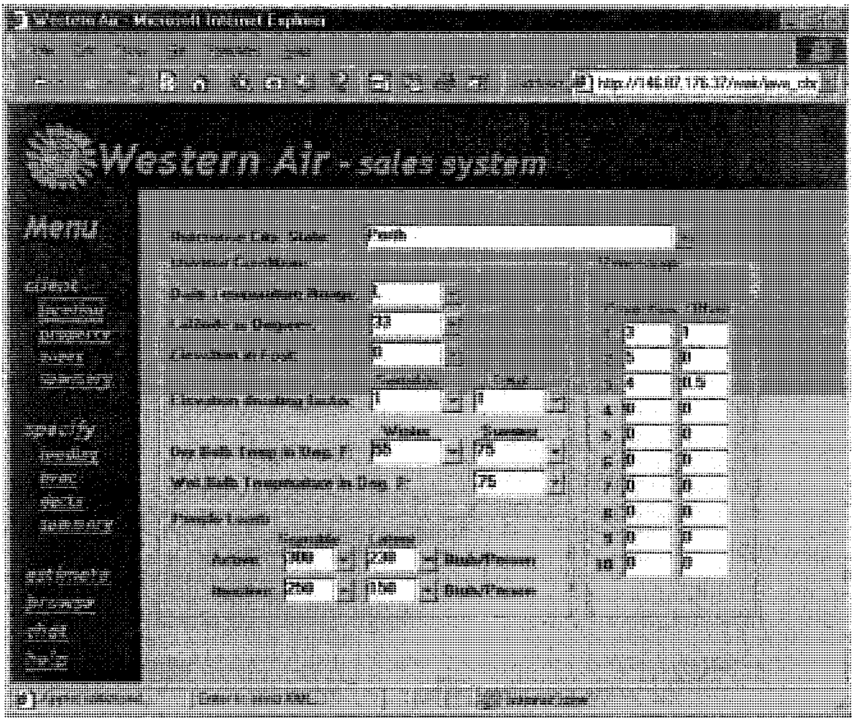


Figure 5. The Java applet showing property location.

4 System Demonstration

Figures 5 to 8 are screen captures showing the systems looks and feel. The first screen (Figure 5) shows part of the capture of the customer's requirements. Figure 6 shows a retrieved case (judged

95% similar) detailing the specification and performance of the HVAC equipment. The subsequent screen shows specification for ducting and finally the last screen shows a summary screen detailing HVAC loads in the customer's living room.

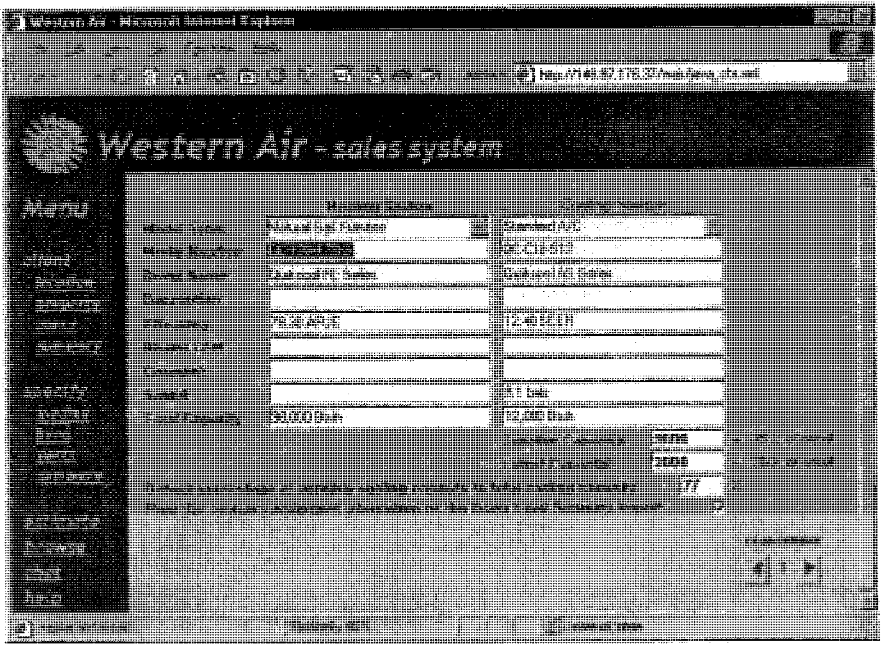
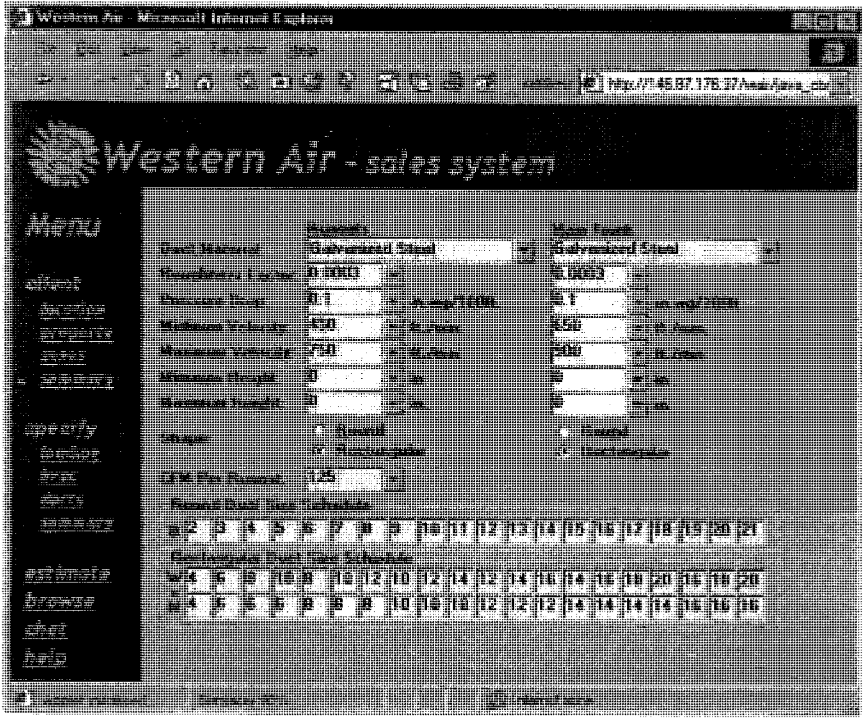


Figure 6. Java applet showing retrieved case HVAC details.

5 Benefits

During the trial month the five sales staff were able to handle 63 installation projects without having an HVAC engineer create the specification. This resulted in a considerable saving in engineers time allowing them more time to deal with complex high value commercial HVAC contracts. It was estimated that margins had been increased by nearly 2% while still remaining competitive. Based on this Western Air has invested $200,000 in purchasing Pentium notebook PCs for its sales staff. The system was rolled out to the entire sales staff in May of 1998. Western Air are expecting profits to increase by $1 million in the first year directly attributable to this system – a more than reasonable return on the investment of $300,000.



Duct Material: Galvanized Steel | Galvanized Steel

Pressure Drop: 0.1 in. wg/100ft | 0.1 in. wg/100ft

Minimum Velocity: 450 ft./min | 550 ft./min

Maximum Velocity: 750 ft./min | 800 ft./min

Minimum Height: 0 in. | 0 in.

Maximum Height: 0 in. | 0 in.

Shape: ○ Round ● Rectangular | ○ Round ● Rectangular

CFM Per Run: 125

Round Duct Size Schedule

d	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

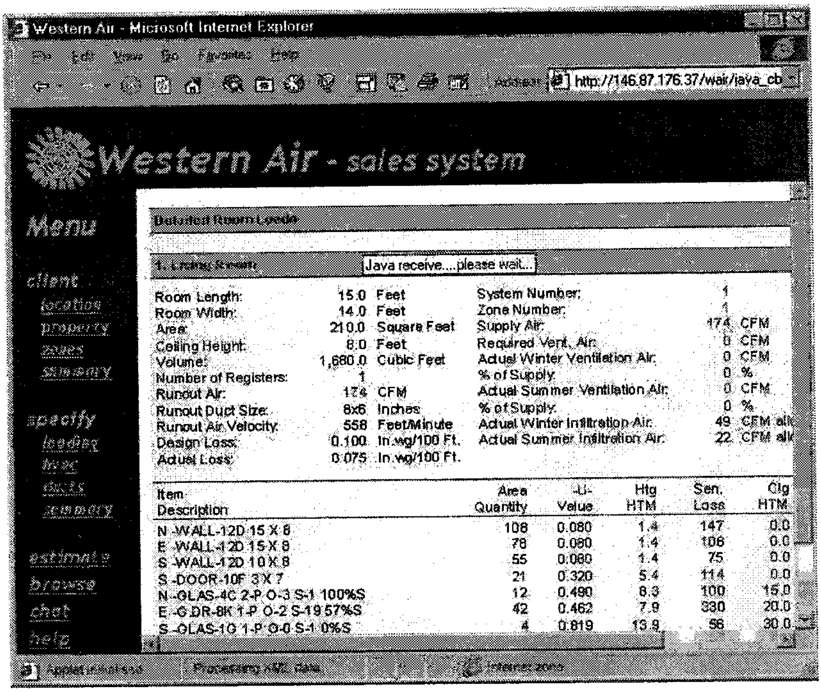
Rectangular Duct Size Schedule

W	4	6	8	10	8	10	12	10	12	14	12	14	16	14	16	18	20	16	18	20
H	4	6	6	6	8	8	8	10	10	10	12	12	12	14	14	14	14	16	16	16

Figure 7. Java applet showing specification of AC ducting.

One of the firm's senior engineers commented that: "*Since this system went live I've had much more time to spend on my own contracts. I used to hate going into the office because I always had a string of problems to handle from the mob out in the field. Now I feel I have the time to really help when I do get a problem to deal with.*"

A member of the sales staff said that: *"This is just great. It used to be really frustrating waiting for them back in Fremantle to deal* *with our problems. I always had to give 'em aggro and when we did finally get an answer the bloody customer changed his mind. Then they whinge because we can't give them an answer on the spot. Now I can even use their phone and get good answers real quick. It really impresses them!"*


Western Air - Microsoft Internet Explorer
http://146.87.176.37/wair/java_cb
Western Air - sales system
Menu
client
location
property
zones
summary
specify
loading
hvac
ducts
summary
estimate
browse
chat
help
Java receive....please wait..

Room Length:	15.0	Feet	System Number:	1	
Room Width:	14.0	Feet	Zone Number:	1	
Area:	210.0	Square Feet	Supply Air:	174	CFM
Ceiling Height:	8.0	Feet	Required Vent. Air:	0	CFM
Volume:	1,680.0	Cubic Feet	Actual Winter Ventilation Air:	0	CFM
Number of Registers:	1		% of Supply:	0	%
Runout Air:	174	CFM	Actual Summer Ventilation Air:	0	CFM
Runout Duct Size:	8x6	Inches	% of Supply:	0	%
Runout Air Velocity:	558	Feet/Minute	Actual Winter Infiltration Air:	49	CFM all
Design Loss:	0.100	In.wg/100 Ft.	Actual Summer Infiltration Air:	22	CFM all
Actual Loss:	0.075	In.wg/100 Ft.			

Item Description	Area Quantity	-U- Value	Htg HTM	Sen. Loss	Clg HTM
N -WALL-12D 15 X 8	108	0.080	1.4	147	0.0
E -WALL-12D 15 X 8	78	0.080	1.4	106	0.0
S -WALL-12D 10 X 8	55	0.080	1.4	75	0.0
S -DOOR-10F 3 X 7	21	0.320	5.4	114	0.0
N -GLAS-4C 2-P O-3 S-1 100%S	12	0.490	8.3	100	15.0
E -G DR-8K 1-P O-2 S-19 57%S	42	0.462	7.9	330	20.0
S -GLAS-1G 1-P O-0 S-1 0%S	4	0.819	13.9	56	30.0

Figure 8. Java Applet showing summary of room HVAC loading.

6 Maintenance

In some sense a CBR system is never finished. The retain process of the CBR-cycle (Aamodt and Plaza 1994) means that the case-base is constantly growing. The concept of completeness in the knowledge-based systems literature only applies in domains where the domain theory or model is well understood. CBR systems most often operate in weak theory domains and the case-base could only be complete if all possible problems in the domain were covered. This is very rarely the situation.

Moreover, although the problem domain must be reasonably stable for similar problems to repeat and hence CBR to be useful (Kolodner 1996), the world does change. There are both explicit and implicit changes in the reasoning environment and problem focus, which will influence the fit of the case-base to the problem context. This will affect the quality and efficiency of the system's results. Thus, it has been recognized for some time now within the CBR community that to keep a system's performance at acceptable levels routine maintenance will be required (Watson 1997, Leake and Wilson 1998, Smyth 1998).

The remainder of the chapter explores the case-base maintenance (CBM) issues encountered after two years of operational use of the system. These were case redundancy caused by the acquisition of many functionally similar cases and case obsolescence primarily caused by equipment obsolescence. I describe the issues encountered and the remedies applied and shows how initial design issues have affected how CBM is required. I then discuss the maintenance required by the system in terms of Richter's knowledge containers (1995) and Leake and Wilson's CBM framework (1998). I conclude by noting that there may be a relationship between sophistication or complexity of the case-representation, similarity metrics and retrieval algorithms and the complexity of CBM. This relationship has software engineering implications; effort put into design may reduce CBM, but will of course increase the initial cost of the system.

7 Case-Base Usage and Growth

The company realized from the outset that use of Cool Air (as the system had become known) could not be optional. To ensure consistency all sales engineers had to use the system. This was ensured by making the system useful even to experienced engineers who did not believe they needed assistance in specifying projects. Thus, from the initial design Cool Air had a form-filling role, work that had to done to record and process a quotation anyway for company records. In a sense, the case-based assistance was a bonus. Consequently, Cool Air was widely used from the start.

In May of 1998, the database contained approximately 10,000 records. These were all relatively recent HVAC installations dating back no more than 5 years. Projects were not consistently stored in a digital format until the mid 1990s.

The company employs approximately 100 sales engineers each of whom deal with an average of five quotations a week (this average is a little misleading since project size and complexity varies greatly from simple residential systems to complex retail and commercial systems). Engineers work for 48 weeks in the year and so the company generates about 24,000 specifications and quotations a year. The company expects to win about 25% of the tenders (i.e., 6,000 installations). Of these from 10 to 20% will not proceed because the customer will change their mind for some reason. Thus, the company expects to perform about 5,000 HVAC installations per year. Actual figures are shown in Table 1.

Table 1. Number of HVAC installations by year.

year	no. installations
1998 (may – dec)	2633
1999 (jan – dec)	5174
2000 (jan – may)	1984
total	9791

All successfully completed installations are retained in the casebase. Moreover, any installation problems are recorded and stored enabling lessons learned from installations to be captured and shown to engineers in future (Watson, 2000).

The number of installations is therefore directly equivalent to the number of new cases retained by Cool Air. Thus, Cool Air's case-base has practically doubled in two years (from 10,000 to 19,791 cases). This considerable growth raised concerns about the utility problem with respect to case retrieval (Smyth and Cunningham 1996) and suggested that a case deletion technique would be required to control the case-base growth (Smyth and Keane 1995).

8 Maintenance Issues

This section describes how two CBM issues were dealt with; namely, functionally redundant cases and obsolete cases. Several general system problems requiring maintenance other than to the case-base were also found and are reported in (Watson and Gardingen 1999).

8.1 Functionally Redundant Cases

Many HVAC installations are very similar, even identical to each other. For example, within a new housing development there may be several identical house designs repeated throughout the development. Moreover, a developer frequently builds identical properties in different locations. Thus, within the case-base there are many functionally identical cases with different location and client details.

Cool Air has a two stage retrieval process. In the first server side process a set of similar cases (approx. 20) is retrieved from the database and sent to the client-side applet. Clearly, there is no point is sending 20 identical cases where one would suffice.

Three solutions to this problem were considered:

1. Just send one case to the client when all cases in the retrieved set are identical. This was rejected because the servlet does not

know that the cases have the same similarity measure. The SQL query retrieves a set that matches within defined limits, the production of a numeric similarity metric is done by the client side applet. Moreover, even if this were possible, it is undesirable because the sales engineers want to be presented with a set of alternatives from which they choose and create a solution. They do not want to be given a single solution.

2. Change the retrieval algorithm on the server side so that it could measure similarity, reject identical redundant cases and construct a useful set of alternatives to send to the client applet. This was rejected because it would have meant completely changing the server side algorithm, which was felt to be working fine. Moreover, it didn't confront the problem of the presence of functionally redundant cases in the case-base.
3. Examine the case-base, identify and remove functionally redundant cases.

Option 3 was chosen as being the sensible solution. There were three alternative solutions:

1. Automatic – an algorithm would be designed to analyze the case-base and automatically identify and remove redundant cases. This algorithm could be run periodically (perhaps weekly) to remove redundancy.
2. Manual – someone would periodically examine the case-base, identify and remove redundant records.
3. Semi-automatic – an algorithm would analyze the case-base and automatically identify sets or clusters of similar cases, flag these and a person would select one case from the set to represent it; the others would be archived.

Solution 2 was rejected because the task would be difficult and tedious to perform manually. Solution 3 was chosen, at least initially, since its success or failure would help determine if solution 1 was achievable.

8.1.1 Redundancy Algorithm Design

Each record in the database contains a field to reference installations that were part of a larger development, such as a housing, apartment or retail development. These units within a large development were likely to be similar or even identical. However, this could not be guaranteed since a proportion of multiple unit developments are made up of unique units (this is often used as a selling feature). Moreover, this reference does not identify commonly repeating standard designs used by many developers in many locations. Consequently using an SQL query to simply identify all units within multi-unit development would not solve the problem.

An algorithm had to be developed to inspect the case-base and identify all identical cases. The algorithm (shown in Figure 9) takes each case in turn and compares it to every case in the case-base. Identical cases (or those exceed the similarity threshold t) are added to the case's similarity set. However, if casel is identical to case3, case7 and case9 this results in four identical similarity sets being created each containing cases 1, 3, 7 and 9. Consequently, it is necessary to compare all of the similarity sets with each other and delete identical sets.

It was recognized that comparing each case to every other case is not a computationally efficient solution. However, since the algorithm need only be run periodically and can be run off-line overnight or at the weekend, this is unlikely to cause problems in the future. Processing time is much cheaper to the company than consultancy time to improve the algorithmic efficiency.

The Redundant Set Identification (RSI) algorithm shown in Figure 9 outputs a list of similarity sets each containing 6 or more cases that are identical or whose similarity exceed the a predefined similarity threshold. No two sets have the same membership.

```
begin
    for $i$ = 1 to $n$
        for $j$ = 1 to $n$
            if sim($case_i$,$case_j$) >= $t$
                then append($set_i$ ,$case_j$)
                    // add the case to a list
            $j$++
        end
        if length($set_i$) not(> $m$)
            then delete($set_i$)
                // deletes sets with fewer than $m$ members
        $i$++
    end
    for $i$ 1 to $s$
        for $j$ = $i$+1 to $s$
            if $set_i$ == $set_j$
                then delete($set_i$)
                    // deletes the first identical set of the pair
            $j$++
        end
        $i$++
    end
end

  where:  $n$ = number of cases
          $t$ = similarity threshold (default = 1.0)
          $m$ = membership threshold (default = 5)
          $s$ = number or sets created
```

Figure 9. Redundant Set Identification Algorithm.

Initially, the similarity threshold was set to 1.0 (i.e., identical) but the set membership threshold was set to 5. It was felt by engineers that being shown that there were several identical solutions provided a measure of confidence in the solution suggested and there would be 15 or so alternatives available if needed.

Once the sets were identified the system maintainer could examine each set in turn, choose a single case to represent the set and set the status flag of the other members to archive.

8.1.2 Redundancy Algorithm Results

The RSI algorithm was run over the case-base of 19,791 cases. The similarity threshold was set to 1.0 (identical) with the results shown in Table 2.

Table 2. Sets identified with similarity of 1.0.

set membership	<10	<25	<50	<75	<100	<125	<150	≥150	totals
no. sets	11	16	21	17	5	2	5	0	77
no. redundant cases	86	288	777	1122	438	222	655	0	3587

3,587 redundant cases were identified in 77 sets or 18.1 % of the cases were identified as being redundant. This significant percentage was not surprising since if redundant cases were not sufficiently common to be a problem they would not have been noticed by users.

Since cases could be very similar, though not identical, and still be functionally redundant (i.e., there are no significant difference in the HVAC specifications) the similarity threshold was reduced to 0.95 (i.e., 95% similarity). The RSI algorithm gave the following results.

The number or redundant cases identified had increased to 5,427 (27.4% of the case-base) the number of similarity sets only increased by 11, whilst the number of cases increased by 1,840. This is because with the weaker similarity threshold more cases are being added to existing similarity sets. If the similarity threshold were set to zero all cases would belong to a single similarity set.

8.1.3 Selecting a Set Representative

Once the similarity sets were identified, the next task was to examine each set and select a single case to represent it. The remaining cases in the set would have their status flag set to archive and thus be ignored in future case-base retrievals. Three strategies were considered:

1. manually select the representative,
2. randomly select the representative,
3. select the median case, i.e., the case with the greatest similarity to all cases in the set.

Solution 1 was rejected because the engineer selected to perform the task said that they found it difficult to decide and admitted to randomly selecting a "likely looking candidate." In effect little different from solution 2. A simple algorithm was written to select the median cases as shown in Figure 10.

This algorithm creates a list containing the representative case from each similarity set (i.e., the case with the highest total similarity to other cases in its set, in the event of several cases having an equal highest total similarity the first case was selected). These cases are retained while all other cases in the similarity sets have their status flags set to archive.

```
begin
    for $i$ = 0 to $s$
        $n$ = length(set$_i$)
        for $j$ = 1 to $n$
            sim$_j$ = sim(case$_i$, element(set$_i$, 1-$n$)) – 1
                                // calc similarity to other cases
            append(simList , sim$_j$)   // add that total to the list
            $j$++
        end
        sort(simList)       // sort the list by descending order
        append(caseList, element(simList, 1))
                                // add the first element to the case list
        $i$++
    end
end

  where: $s$ = number of similarity sets
```

Figure 10. Median Case Identification Algorithm.

The application of this algorithm reduced the case-base by 5329 cases (i.e., 5427 cases less one representative from 98 similarity sets) as shown in Table 3. The new case-base contained 14,462 cases, which still represents a significant increase in case-base size from its original size.

Table 3. Sets identified with similarity of 0.95.

set membership	<10	<25	<50	<75	<100	<125	<150	≥150	totals
no. clusters	15	19	25	21	7	4	6	1	98
no. redundant cases	135	437	1153	1512	665	487	882	156	5427

8.2 Functionally Obsolete Cases

The second CBM issue related to case obsolescence. Over time, HVAC equipment is withdrawn and replaced and working practices change. Cases referring to installations using obsolete products or techniques need to be deleted from the case-base to prevent inexperienced engineers including them in new specifications and quotes. The company releases weekly technical memoranda by email and specific working practice guidelines, which are, updated quarterly. Moreover, sales engineers receive twice annual training to ensure they up to date with current products and practice.

Some CBR systems retain details of obsolete cases since these maybe provide useful analogies for problem solving in future. It is common for trouble-shooting or diagnostic case-bases to retain cases referring to problems with obsolete equipment because similar problems may occur in future with new equipment (Klahr 1996). However, it was decided by management that installations using obsolete equipment need not be retained for problem solving in Cool Air.

It was a relatively easy administrative job to search the database to identify and archive cases that refer to obsolete equipment so they are not included in the case-base retrieval process. This is done each time there is a significant product change. However, changes also need to be made to the symbol hierarchies used by the SQL query relaxation technique. This was not anticipated during the design of the system. Editing the symbol hierarchies (a sample is shown in Figure 11) to remove obsolete items of equipment or entire classes of equipment is not simple. They are stored as tables within the database and a good knowledge of the table structure and relations between them is required to ensure that the hierarchy is not corrupted.

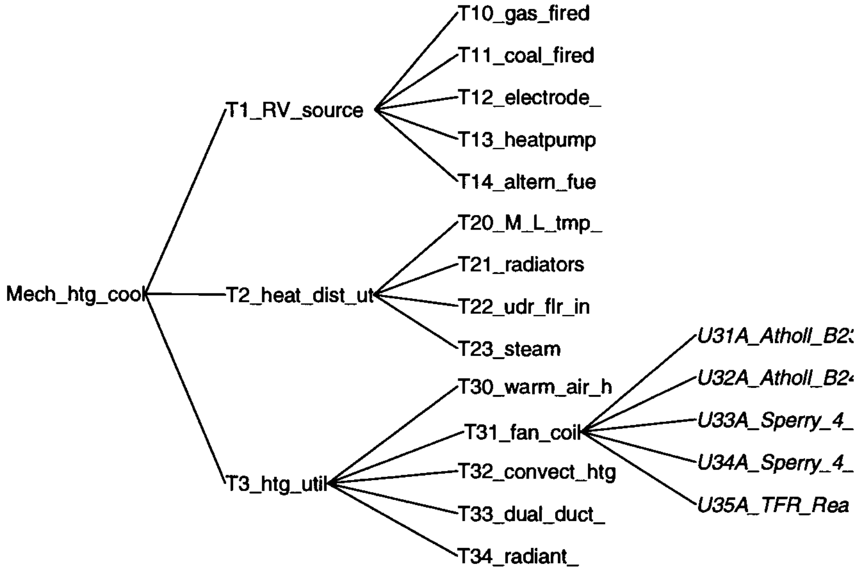


Figure 11. A portion of the symbol hierarchy for mechanical heating & cooling systems.

It is not clear that this can be done automatically or even semi-automatically. A graphical hierarchy editor would greatly help the editing task and make it more feasible for a domain expert to do the maintenance rather than a programmer. This has been suggested to the company but is currently beyond their budget for the system.

Finally, it remains unclear how to identify records where obsolete working practices were used since these are not explicitly referred to in the record structure but remain hidden in the supporting files on the FTP server (see Figure 2) or are not even recoded at all. Working practices were not considered as important during the design of either the database or the CBR system and this is an ongoing issue that has not been resolved.

It is worth comparing this with the experience of Kitano and Shimazu (1996) at NEC with SQUAD. Cool Air's query relaxation SQL technique is derived from that of the SQUAD system, but beyond this, the system's differ significantly. The SQUAD system was design as a corporate memory of software quality control problems. Whilst it contains over 20,000 cases, it is not clear to what degree redundancy or obsolescence of cases is or was a problem. However, Kitano and Shimazu (1996) do state that using a DBMS to manage a large case-base is extremely useful.

9 Discussion

The two years of use of Cool Air has provided an interesting case study of a commercially fielded CBR system. CBM was recognized as an issue during the initial design and management were aware that the system would require regular maintenance if it were to remain useful.

It is useful to analyze the maintenance the system has required in terms of Richter's knowledge containers terminology (Richter 1995). Since as Richter noted in theory each of the knowledge containers of a CBR system may require maintenance as the domain knowledge changes subtly and the case-base grows.

1. *vocabulary* – Cool Air has the same case features now as it did two years ago, no vocabulary maintenance has been done. However, use of the system has shown that we failed to capture

knowledge about working practice or methods within the case vocabulary and this should be addressed in the future.

2. *similarity measure* – the concept hierarchy is an essential component of the similarity-based retrieval from the database. This has required maintenance. However, other components within the system's similarity measure have not required maintenance. It would be unwise to generalize from this since Cool Air is a very relaxed system in that it need only retrieve good or likely candidates. A CBR system that had to retrieve with greater precision may well require maintenance in this area.
3. *case-base* – this has required extensive maintenance that is repeated periodically. Two forms of CBM were carried out: (1) redundant cases were identified and removed using an introspective reasoning technique approximately on a monthly cycle and (2) obsolete cases were removed using standard DBMS tools whenever a significant obsolescence is identified. It is worth remembering here that DBMS techniques were not sufficient to identify redundant cases because redundancy is dependent on similarity.
4. *solution transformation* – Cool Air does not perform adaptation, this is left to the engineers so no maintenance was required here.

It can be argued that the redundancy problems we encountered were due to (or exacerbated by) the design of Cool Air and the two-stage retrieval process in particular. Although we have not observed the utility problem (i.e., retrieval performance has not suffered), with a case-base doubling in size over two years it would be unwise to ignore the utility problem in the long term.

In comparison to the CBM processes described by Goker and Roth-Berghofer (1999) for the HOMER system it must be recognized that Western Air is a very small company in comparison to DaimlerChrysler and that management processes throughout are much more ad hoc and flexible. Yet, it is clear that like HOMER Cool Air does follow a distinct "maintenance cycle," although the "Retain" task is done automatically without intervention from a case-base administrator. However, the maintenance of the knowledge containers is undertaken periodically in a "Refine" task and is largely done by introspection using the algorithms described above as recommended for HOMER.

It is also worth analyzing how the maintenance of Cool Air is performed in terms of the case-base maintenance (CBM) framework created by Leake and Wilson (1998). Cool Air's maintenance policy is as follows:

- **Type of data**: *None.* Statistics are gathered on: global usage of the case-base, usage of individual cases and the retention of new cases, but this information is not used to inform either the retention of new cases, when to perform maintenance or what maintenance to perform.
- **Timing**: *Ad hoc & Conditional.* The removal of redundant cases is currently done at irregular intervals. The removal of obsolete cases is conditional on the identification of product obsolescence.
- **Integration**: *Off-line.* Data collection is performed off-line when the system is not being used.
- **Triggering**: *Result-based.* Maintenance will be triggered if a user reports a problem with system
- **Revision level**: *Knowledge-level.* Maintenance focuses on deleting cases.
- **Execution**: *None.* The system executes no maintenance changes itself.
- **Scope of maintenance**: *Broad.* Maintenance operations will typically affect many cases in the case-base. However, if maintenance were scheduled regularly, perhaps after each case retention, then the scope of maintenance would become narrow.

With hindsight, the design of Cool Air should be changed to examine each new case before it is added to the system and only retain it if it is significantly different from other cases already in the case-base (i.e., the case is useful). This would not only be a simpler algorithm to apply than the maintenance algorithms shown in Figures 10 and 11, but would also remove redundancy completely in future. Making this small implementational change would dramatically change Cool Air's maintenance policy to an introspective, continuous, on-line, narrow policy.

10 Lessons Learned

The development of the Cool Air system, like many programs within small companies came about partially because of chance and the vision of one or two people rather than as the product of a carefully managed process. The initial goal of the modest development project as reported in Watson and Gardingen (1999) was to develop a system as quickly and cost effectively as possible. The need to maintain the case-base was made explicit to management from the outset. Yet, CBM become more complex than envisioned. The following lessons can be learned from our experience:

- Case-base developers cannot be certain that they have elicited all correct case features. Whilst not a CBM issue per se developers should plan to revisit the case features some time after the system is rolled out (Note: this is a different issue from the emergence of new case features with time, which is a CBM issue).
- Storing cases in a DBMS is essential if the case-base is of a reasonable size. Most commercial CBR tools now provide this functionality and no longer keep their case in proprietary formats. A DBMS greatly helps with reporting on the case-base, removing or archiving obsolescent cases and with general case-base management.
- A more complex case-representation will be more complex to maintain. However, maintenance may be required less often.
- If developers inherit a case-base from a legacy database, they should consider using an algorithm to analyze the coverage of

the case-base, such as the one proposed by Smith and McKenna (1998). CBM was greatly increased by the presence in the original case-base of many redundant cases. Highlighting this would alert developers to the need to explicitly deal with redundancy at an early stage before even more redundant cases are acquired through new case retention.

- Developers should look at each of Richter's knowledge containers (1995) and inquire what the maintenance needs of each container might be. This would have brought to our attention the need to maintain the symbol hierarchy as equipment changed.

11 Conclusions

Cool Air provides a snap shot of how a web CBR system has been kept in regular profitable use for two years. Although some of the issues dealt with here are specific to this application, the domain and the company it is possible to generalize lessons learned from the case study. Although the designers of Cool Air explicitly considered maintenance from the outset in retrospect this was exclusively from a managerial viewpoint to ensure management commitment to expenditure on regular maintenance. I now realize that had we considered case-base maintenance from an implementational viewpoint during design, with the exception of obsolescence, we could have better designed maintenance into Cool Air. This would have been both more elegant and cost effective. In our defense, I would say that probably in common with most CBR system implementers we were so focused on "getting retrieval to work" that maintenance became a "we'll deal with that later" issue. Designers of CBR systems should therefore remember that they need to design a system from the outset that deals with all of the processes of the CBR-Cycle and not just the retrieval, reuse and revision processes.

There is however, an important software engineering issue here. Developers of CBR systems will always be under pressure to deliver a functional CBR system quickly. One of the selling points of CBR systems has always been that that they can be implemented quickly (Harmon 1992). We have recognized that more complex case-representations, similarity metrics and retrieval algorithms can improve the precision and/or efficiency of retrieval. However, this complexity increases the cost of developing systems. This trade off is also true for CBM. A richer case representation would have partially solved the redundancy problem identified in Section 8.1.1. More extensive knowledge engineering might have captured that working practices should have been a case feature, as noted in Section 8.2. Both of these would have increased the development cost, have reduced the return on investment and may have resulted in the system never getting the management backing required to deliver it. A great strength of CBR is it's simplicity – the symbol hierarchy used by the system to assess similarity (see Figure 11) is already an example where the complexity of the system has exceeded the ability of it's end users to maintain it without professional programming assistance.

Developers of commercial CBR systems are advised to keep their system simple but to recognize that there may be relationship between sophistication or complexity of the CBR system and the complexity of CBM. This relationship has software engineering implications; effort put into design, representation and architecture may reduce CBM, but may increase the complexity and initial cost of the system. Moreover, although a more complex system might require less CBM (perhaps because more routine tasks are automated) that which is required may need the skills of a specialist knowledge engineer or programmer.

For the future, unfortunately, it is not at all clear how Cool Air will develop or to what degree it will be professionally and properly maintained. Many users within the company view the system as no more than a database and do not appreciate its sophistication. Only a few employees are aware that without regular CBM Cool Air's performance will steadily degrade. It is also unlikely that the company will develop a similar system in the future and so they will not benefit from the lessons learned.

References

Aamodt, E. and Plaza, E. (1994), "Case-based reasoning: foundational issues, methodological variations, and system approaches," *AICom – Artificial Intelligence Communications*, IOS Press, vol. 7, no. 1, pp. 39-59.

Goker, H. and Roth-Berghoffer, T. (1999), "Development and utilization of a case-based help-desk support system in a corporate environment," in Althoff *et al.* (eds.), *Case-Based Reasoning Research & Development*, Springer LNAI vol. 1650, pp. 132-146.

Hammond, K.J., Burke, R., and Schmitt, K. (1996), "A case-based approach to knowledge navigation," in Leake, D.B. (ed.), *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, AAAI Press/The MIT Press Menlo Park, Calif., US, pp. 125-136.

Harmon, P. (1992), "Case-based reasoning III," *Intelligent Software Strategies*, vol. 8, no. 1.

Kitano, H. and Shimazu, H. (1996), "The experience sharing architecture: a case study in corporate-wide case-based software quality control," in Leake, D.B. (ed.), *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, AAAI Press/The MIT Press Menlo Park, Calif., US, pp. 235-268.

Klahr, P. (1996), "Global case-based development and deployment," in Smith, I. and Faltings, B. (eds.), *Advances in Case-Based Reasoning*, pp. 392-399, Springer-Verlag LNAI vol. 1168, pp. 519-530.

Kolodner, J.L. (1996), "Making the implicit explicit: clarifying the principles of case-based reasoning," in Leake, D.B. (ed.), *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, AAAI Press/The MIT Press, Menlo Park, Calif., USA.

Leake, D.B. and Wilson, D.C. (1998), "Categorizing case-base maintenance: dimensions and directions," in Smyth, B. and Cunningham, P. (eds.), *Advances in Case-Based Reasoning*, Springer Verlag LNAI vol. 1488, pp. 196-207.

Richter, M. (1995), "The knowledge contained in similarity measures," Invited talk at *ICCBR'95*. Available online at http:// www.cbr-web.org/documents/Richtericcbr95remarks.html .

Richter, M. (1998), "Introduction – the basic concepts of CBR," in Lenz, M., Bartsch-Sporl, B., Burkhard. H.-D., and Wess, S. (eds.), *Case-Based Reasoning Technology: from Foundations to Applications*, LNAI vol. 1400, Springer-Verlag, Berlin.

Sengupta, A., Wilson, D.C., and Leake, D.B. (1999), "On constructing the right sort of CBR implementation," *Proc. of the IJCAI-99 Workshop on Automating the Construction of Case Based Reasoners*, Stockholm, Sweden.

Shimazu, H. (1998), "Textual case-based reasoning system using XML on the World-Wide Web," *Proc. of the 4th European Workshop on CBR (EWCBR98)*, Springer Verlag LNAI.

Smyth, B. (1998), "Case-base maintenance," *Proc. of the 11th. International Conference on Industrial & Engineering Applications of AI and Expert Systems.*

Smyth, B. and Cunningham, P. (1996), "The utility problem analysed: a case-based reasoning perspective," in Smith, I. and Faltings, B. (eds.), *Advances in Case-Based Reasoning*, Springer-Verlag LNAI vol. 1168, pp. 392-399.

Smyth, B. and Keane, M. (1995), "Remembering to forget: a competence-preserving case deletion policy for case-based reasoning systems," *Proc. 13th International Joint Conference on Artificial Intelligence*, pp. 377-382.

Smyth, B. and McKenna, E. (1998), "Modelling the competence of case-bases," in Smyth, B. and Cunningham, P. (eds.), *Advances in Case-Based Reasoning*, Springer Verlag LNAI vol. 1488, pp. 208-220.

Watson, I. (1997), *Applying Case-Based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufmann Publishers Inc., San Francisco, CA.

Watson, I. (2000), "A case-based reasoning application for engineering sales support using introspective reasoning," *Proc. 17th. National Conference on Artificial Intelligence (AAA!-2000) and the 12th Innovative Applications of Artificial Intelligence Conference (IAAI-2000)*, July 30 – August 3, Austin Texas, pp. 1054-1059. AAAI Press.

Watson, I. and Gardingen, D. (1999), "A distributed case-based reasoning application for engineering sales support," *Proc. 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99)*, vol. 1, pp. 600-605. Morgan Kaufmann Publishers Inc.

World Wide Web Consortium (1997), "Extensible Markup Language 1.0, recommendation by W3C." Available online at http://www.w3.org/TR/PR-xml-971208 .

This page is intentionally left blank

Index

List of Contributors

S. Azechi
School of International Cultural Relations
Hokkaido Tokai University
Sapporo 005-8601
Japan
s_azechi@di.htokai.ac.jp

W. Balzano
Dipartimento di Informatica ed Applicazioni
Università di Salerno
via S. Allende, 84081 Baronissi (SA)
Italy
walbal@dia.unisa.it

S. Botros
Sun Microsystems, Inc.
901 San Antonio Road, Palo Alto, CA 94303-4900
USA

R.W. Brause
J.W.G.-University
Frankfurt a.M.
Germany
Brause@Informatik.Uni-Frankfurt.de

P. Ciancarini
Dipartimento di Scienze dell'Informazione
Università di Bologna
Mura A. Zamboni 7, 40128 Bologna
Italy
cianca@cs.unibo.it

E. Damiani
Dipartimento di Tecnologie dell'Informazione
Università di Milano
Italy
damiani@dti.unimi.it

A. Dattolo
Dipartimento di Matematica ed Applicazioni
Università di Napoli Federico II
via Cinthia, Complesso Universitario di Monte Sant'Angelo,
80126 Napoli
Italy
dattolo@unina.it

N. Fujihara
Research Center for School Education
Naruto University of Education
Naruto 772-8502
Japan
fujihara@naruto-u.ac.jp

T. Fukuhara
Synsophy Project
Communications Research Laboratory
Kyoto 619-0289
Japan
tomohi-f@synsophy.go.jp
and
Graduate School of Information Science
Nara Institute of Science and Technology
Nara 630-0101
Japan

T. Ichimura
Faculty of Information Sciences
Hiroshima City University
Japan

L. Ingber
DUNN Capital Management
309 E Osceola St, Ste 208, Stuart, FL 34994
ingber@ingber.com, ingber@alumni.caltech.edu

H. Kubota
School of Engineering
The University of Tokyo
Tokyo 113-8656
Japan
nishida@kc.t.u-tokyo.ac.jp

N. Lavarini
Student Placement
BT Research
UK
lavarn@info.bt.co.uk

R.S.T. Lee
Department of Computing
Hong Kong Polytechnic University
Hung Hom
Hong Kong

S. Marrara
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Italy
marrara@elet.polimi.it

K. Mera
Faculty of Information Sciences
Hiroshima City University
Japan

R.P. Mondescu
DRW Investments LLC
311 S Wacker Dr, Ste 900, Chicago, IL 60606
rmondescu@drwtrading.com

T. Nishida
School of Engineering
The University of Tokyo
Tokyo 113-8656
Japan
nishida@kc.t.u-tokyo.ac.jp

B. Oliboni
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Italy
oliboni@elet.polimi.it

R. Rizzo
Italian National Research Council
Institute for Educational and Training Technologies
Italy
rizzo@itdf.pa.cnr.it

A. Sato
Institute of Information Sciences and Electronics
University of Tsukuba
Ibaraki
Japan

N. Shirahama
Kitakyushu National College of Technology
Fukuoka
Japan

L. Tanca
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Italy
tanca@elet.polimi.it

M. Ueberall
J.W.G.-University
Frankfurt a.M.
Germany
Markus@Informatik.Uni-Frankfurt.de

F. Vitali
Dipartimento di Scienze dell'Informazione
Università di Bologna
Mura A. Zamboni 7, 40128 Bologna
Italy
fabio@cs.unibo.it

S. Waterhouse
Sun Microsystems, Inc.
901 San Antonio Road, Palo Alto, CA 94303-4900
USA

I. Watson
AI-CBR
Department of Computer Science
University of Auckland
Auckland
New Zealand
ian@ai-cbr.org

T. Yamashita
Graduate School of Engineering
Tokyo Metropolitan Institute of Technology
Tokyo
Japan

The internet/WWW has made it possible to easily access quantities of information never available before. However, both the amount of information and the variations in quality pose obstacles to the efficient use of the medium. Artificial intelligence techniques can be useful tools in this context. Intelligent systems can be applied to searching the internet and data-mining, interpreting internet derived material, the human–Web interface, remote condition monitoring and many other areas.

Internet-Based Intelligent Information Processing Systems

The volume presents the latest research on the interaction between intelligent systems (neural networks, adaptive and connectionist paradigms, fuzzy and rule-based systems, intelligent agents) and the Internet/WWW. It surveys both the employment of intelligent systems to facilitate and enhance the use of the Internet, and applications where the Internet is a channel through which intelligent techniques are applied.

World Scientific
www.worldscientific.com
5204 hc

ISBN 981-238-281-X
9 789812 382818