

Architectura et Machina

Architectura et Machina

Computer Aided Architectural Design und Virtuelle Architektur

Gerhard Schmitt



Prof. Dr. Gerhard N. Schmitt
Professur für Architektur und CAAD
Abteilung für Architektur
Eidgenössische Technische Hochschule
ETH Zürich, CH-8093 Zürich

Alle Rechte vorbehalten

© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1993
Softcover reprint of the hardcover 1st edition 1993

Der Verlag Vieweg ist ein Unternehmen der Verlagsgruppe Bertelsmann International.



Das Werk einschließlich aller seiner Zeilen ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urhebergesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigung, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Layout und Satz: Florian Wenz

Einband: Florian Wenz, nach einem Bild von Zoran Sladojew

Inhalt

Einleitung	8
Dank	10
 <i>Architectura</i>	
CAAD und Geschichte - computus et historia (Werner Oechslin)	14
Architectura - eine Wissenschaft	24
Traditionelle Entwurfsprozesse - Medium Papier	25
Wissenschaftliche Entwurfsprozesse - Design Methods	26
Neue Entwurfsprozesse - Medium Computer	30
Repräsentation von Architektur	32
Methoden für den computerunterstützten Entwurf	36
Abstraktion und Modellbildung	38
Simulation	40
Top-Down	42
Bottom-Up	44
Prototypen	46
Fallbasiertes Schließen	48
Maschinenlernen	50
Instrumente für den computerunterstützten Entwurf	52
Editieren I - Zahlen und Text	54
Editieren II - Diagramme	56
Editieren III - Geometrie	58
Programmieren I - Traditionelle Programme	60
Programmieren II - Wissensbasierte Programme	62
Programmieren III - Objektorientierte Programme	64

Generieren I - Formengrammatiken	66
Generieren II - Fraktale	68
Parametrisierung	70
Objektorientiertes Modellieren I - Typen und Variationen	72
Objektorientiertes Modellieren II - Substitution	74
Objektorientiertes Modellieren III - Detaillierungsgrade	76
Objektorientiertes Modellieren IV - Designfokus - Logical Zoom	78
Zusammenfassung	80

Et

Et - Anwendungen in Lehre, Praxis und Forschung	84
Lehre und Machina	85
Das neue Instrumentarium	89
Ordnung und Chaos	91
Entwurfshilfen	93
Integrierter Entwurf	95
Modelle und Präsentation	99
Principia: Entwurf als Sprache	104
Schritt 1: Vokabular und erste Kompositionen auf der Ebene	106
Schritt 2: Konstruktion im Raum und Substitution	108
Schritt 3: Multiple Repräsentation und Detaillierungsgrad	110
Schritt 4: Hierarchien und Compound Objects	112
Principia: Objektorientiertes Entwerfen	114
Principia: Modellieren mit Prototypen	118
Rekonstruktion einer antiken Stadt: Aventicum	120
Umnutzung eines Industrieareals: Winterthur	126
Rekonstruktion und Analyse der Architektur Giuseppe Terragni (Antonino Saggi)	128
CAAD-Programmentwicklung	132
Praxis und Machina	140
Ein Wettbewerb für Granada (Leandro Madrazo)	142
Projekt: Simulation eines neuen Chemiegebäudes	146
Ausstellung: New Realities	148
Ausstellung: Lineamenta-CAAD (Leandro Madrazo)	150
Projekt: Der Traum vom großen blauen Wasser	158
Projekt: Triemli	160
Projekt: Rote Drachenstele (Urs Roth)	162
Projekt: Kraftstrasse 35 (Florian Wenz)	168
Forschung und Machina	172

Wissenschaftliche Rekonstruktion: Eileen Grays Maison en bord de mer (Stefan Hecker)	175
Geodatenverarbeitung	178
Fallbasiertes Entwerfen	184
Intelligente Objekte	191
Virtuelle Realität in der Architektur	193

Machina

Machina - die Maschine	204
<i>Die Maschine in der Gesellschaft: Materielle und intellektuelle Infrastruktur</i>	205
Die Maschine - Hardware	205
Das Betriebssystem	207
Vernetzung - Networks	207
Computersprachen - Computer Languages	208
Programme - Software	208
Benutzeroberflächen - User Interfaces	209
<i>Die Maschine im Architekturbüro</i>	211
Grundlagen des CAD	211
Neue Chancen für das Architekturbüro	212
Gesichtspunkte bei der Anschaffung eines CAD- Systems	213
Die Schulung der Mitarbeiterinnen und Mitarbeiter	213
Ökonomische Aspekte - Kosten und Nutzen	214
<i>Die neuen Instrumente der Architektur: CAAD-Komponenten und Programme</i>	216
Hardware für die Architektur	216
Programme für die Konstruktion	216
Datenbanken	217
Integrierte Programme	218
Das elektronische Skizzenbuch	219
Rendering und Animation	219
Ein CAAD-System	220
<i>Maschine und Künstliche Intelligenz (KI)</i>	222
Architektur und KI	223
<i>Die nächste Generation</i>	224
Auswirkungen der Vernetzung	224
<i>Rückblick und Ausblick - eine mögliche Zukunft</i>	226
<i>Glossar</i>	228
<i>Bibliographie</i>	236
<i>Stichwortverzeichnis</i>	244

Einleitung

Computer Aided Architectural Design und Virtuelle Architektur

«Architectura et machina» ist der erste Teil einer auf drei Bände angelegten Arbeit, deren zweiter Titel «Architectura cum machina» und deren dritter Titel «Architectura ex machina» heißen wird. Damit sind bereits der Startpunkt und das vorläufige Ziel einer Entwicklung definiert, deren historische Anfänge im 14. Jahrhundert liegen und deren aktivste Phase gerade bevorsteht. Noch können Architektinnen und Architekten das Ergebnis dieser Entwicklung wesentlich mitbestimmen. «Architectura et machina» ist die Aufforderung und eine Anleitung dazu, das faszinierende Gebiet der zukünftigen Architekturmethoden und Architekturinstrumente zu definieren.

«Architectura et machina» handelt vom computerunterstützten architektonischen Entwerfen, international bekannt als Computer Aided Architectural Design oder CAAD. Noch gibt es keine selbst entwerfenden Computerprogramme, deshalb die Konjunktion «et» im Titel. Bisher stehen traditionelle Architektinnen und Architekten sowie solche, die CAAD bewußt im Entwurfsprozeß einsetzen, einander noch isoliert gegenüber - Welten, die zwischen ihnen liegen. Das traditionelle Entwerfen hat einen Endpunkt erreicht und sucht nach neuen Möglichkeiten, CAAD eröffnet eine New Frontier. Vereinfacht gesehen, birgt das traditionelle Entwerfen einen großen Schatz an Erfahrung, aber wenig Visionen, wogegen CAAD kaum Erfahrung, aber weitreichende Visionen mitbringt. Die Absicht des Buches ist es nicht, bestehende Entwurfs-, Modellier- und Zeichenmethoden unverändert auf den Computer zu übertragen, sondern neue Möglichkeiten der Maschine zu erkunden und sie mit Hilfe von Experiment und Test für die Architektur nutzbar zu machen. «Architectura et machina» propagiert also keine maschinengerechte Architektur, sondern eine architekturgerechte Maschine.

Die Architektur, die Maschine oder der Computer sind keine sich selbst definierenden Größen, sie haben kein eigenes Leben und es ist für die wissenschaftliche Betrachtung sehr hinderlich, sie zu personifizieren. Architektur und Computer werden von Menschen entworfen und gebaut. Das Entwerfen und das Bauen manifestieren eine Geisteshaltung und eine Kombination von Möglichkeiten und Beschränkungen, die zur jeweiligen Form des Objektes führen. Dies darf während der Lektüre des Buches nicht vergessen werden. Computer haben sich allerdings bereits zu Projektionsmaschinen entwickelt. Das bedeutet, daß eine große Zahl von Anwendern Wünsche, Verhaltens- und Vorgehensweisen auf die Maschine projiziert und diese dann dort bestätigt oder widerlegt findet.

An den Anfang gehören einige grundsätzliche Positionen zum Thema CAAD. Die Befürworter erwarten sich von der Einführung des CAAD die Lösung von Problemen in der Administration, im Entwurf, in der Dokumentationsphase, in der Bauabwicklung und besonders im Gebäudemanagement. An den Universitäten setzen die Forscherinnen und Forscher besonders auf eine Verbesserung des Entwurfsprozesses und der resultierenden Architektur. Die Gegner des CAAD weisen auf die bisher fehlenden positiven Resultate hin. Ihre Argumentation reicht von völliger Ablehnung von Computern im Entwurf bis zu dem versuchten Nachweis, daß Computer nie entwerfen können werden. Selbst die Anwender von Computern in der Architektur sind sich nicht einig über die Bedeutung und die Stellung der neuen Methoden und Instrumente für ihre Arbeit. In dieser Diskussion stehen sich die Vertreter des strukturierten und rationalen Vorgehens auf der einen, sowie die Verfechter der interaktiven, künstlerischen Anwendung des Computers auf der anderen Seite gegenüber. Die Frage taucht dabei immer wieder auf, ob das, was wir heute

als speziell menschliche Fähigkeit einschätzen, in Zukunft von Maschinen simuliert werden darf, oder ob der Computer lediglich Rechen-, Schreib- und Zeichenmaschine bleiben soll. Computer sind schneller und exakter als Menschen, wenn es sich um Kalkulationen mit einer großen Zahl von iterativen oder rekursiven Schritten handelt, also um Syntax. Beispiele sind die Berechnung verdeckter Kanten in Perspektiven, Auswerten von Kostenalternativen, oder die Simulation des Energiebedarfs. Sie sind noch schwächer auf Gebieten, die auf der semantischen Interpretation hauptsächlich graphischer Objekte beruhen: das direkte und sofortige Erkennen von wichtigen Umrissen oder Details in Umgebungen mit viel überflüssiger Information, das Interpretieren eines Grundrisses, die blitzschnelle Beurteilung eines Entwurfs.

An dieser Stelle ist es angebracht, den Begriff 'Computer' zu definieren und dadurch zugleich zu entmystifizieren. Unter Computer verstehen wir eine Maschine (Hardware), die mit Hilfe von Programmen (Software) Informationen verarbeitet. Die Geschwindigkeit der Maschine ist ein wichtiger Faktor, der, wie sich später zeigen wird, viele Operationen erst ermöglicht. Das für CAAD interessante Gebiet ist jedoch die Software. Hier spielen sich die intellektuellen Auseinandersetzungen ab. Selbst die bisher als gegeben vorausgesetzten traditionellen Computersprachen sind im Begriff, sich erheblich zu verändern.

Am Beginn der neunziger Jahre haben Computer im Bereich der Administration, des Schreibens und der quantitativen Analyse ihre Eignung bewiesen. Sie sind zur unentbehrlichen Hilfe bei der Lösung syntaktisch definierbarer Probleme geworden. Die Eignung von Computern für das kreative Umgehen mit den neuen Problemen der Architektur und ihrer Stellung in der Gesellschaft wird sich in der nächsten Zukunft erwei-

sen. «Architectura et machina» ist kein Buch, das der Technikbesessenheit, der Machbarkeit, Berechenbarkeit und Meßbarkeit oder gar der neuen Mystik einer kommenden Architektur das Wort redet. Es geht vielmehr von der Überzeugung aus, daß alles neue Wissen und alle neuen Instrumente eingesetzt werden müssen, um eine verantwortlichere Architektur zu entwerfen und zu bauen. Verantwortung bezieht sich auf den Schutz von Leben, Umwelt und ästhetischem Gefühl. Die Erfahrung der Vergangenheit kann uns dabei nur beschränkt helfen, da die Gestaltung der Zukunft anderes Wissen, Willen und Mut verlangt.

Architektur ist eine faszinierende Disziplin, in der ständig neue Instrumente zum Einsatz kommen. Das Resultat war, ist und wird Architektur sein. Die Maschine (machina) hat in der Form des Computers einen Entwicklungsstand erreicht, der sie zum nützlichen intellektuellen Instrument des Menschen macht. Die Weiterentwicklung dieser Maschine in Form von Hardware und Software ist zur Zeit die dynamischste Erscheinung in den nachindustriellen Gesellschaften. Beide können voneinander profitieren: die Architektur von den neuen Möglichkeiten der Maschine, die Maschine von der Entwurfstradition der Architektur. Den Architektinnen und Architekten des ausgehenden 20. Jahrhunderts sind beide bekannt. «Architectura», der erste Teil des Buches, und «machina», der letzte Teil des Buches, bilden so zwei Pole, zwischen denen sich interessante Anwendungen entwickeln, die im Mittelteil des Buches unter «et» beschrieben sind. Der Titel ist lateinisch, da dies eine universale Sprache war, mit der eine Disziplinübergreifende Verständigung möglich wurde. Architektur und Computer erheben berechtigterweise ebenfalls einen gewissen Anspruch auf Universalität.

Dank

«Architectura et machina» stellt Ergebnisse der Arbeit an der Professur für Architektur und CAAD an der ETH Zürich seit ihrer Schaffung im Oktober 1988 dar. Mein Dank gilt deshalb zuerst meinen Mitarbeiterinnen und Mitarbeitern, die mir geholfen haben, die in diesem Buch beschriebenen Ideen zu finden, zu formulieren, umzusetzen und zu testen. Besonders erwähnen möchte ich Maia Engeli, Annelies Zeidler, Sibylla Spycher und Werner Riniker, Chen-Cheng Chen und Shen-Guan Shih aus Taiwan, die mir beim Aufbau in Lehre und Forschung von Anfang an zur Seite standen und die im administrativen und technischen Bereich große Hilfe leisteten; Bharat Dave aus Indien, der mich seit der gemeinsamen Zeit an der Carnegie Mellon University in meiner Arbeit kritisch begleitet und immer wieder neue Forschungsideen einbringt und umsetzt; Nathanea Elte, die mit großer Umsicht im Nachdiplom-Programm arbeitet; Stefan Hecker, der in seinem Beitrag ein interessantes Forschungsprojekt beschreibt; Robert Hupertz, der große praktische Erfahrung in ein Forschungsprojekt einbrachte; David Kurmann, der der erste Nicht-Architekt im Team war und wichtige Beiträge auf der Seite der Informatik und neuer Benutzeroberflächen leistete; Laura Lee aus Kanada, die in den ersten Semestern erfolgreich die Lehre mitgestaltete und ihre Arbeit an der Carnegie Mellon University fortsetzte; besonderer Dank gebührt Leandro Madrazo aus Spanien, der über erfolgreiche Arbeiten an der Harvard University und bei Skidmore, Owings, & Merrill in Los Angeles zu uns stieß und der viele neue Ideen in der Lehre entwickelte und mit den Studierenden realisierte; von ihm stammen Textbeiträge und zahlreiche Anregungen für dieses Buch sowie die ersten wichtigen Umsetzungen der theoretischen Methoden und Instrumente in Entwürfe und Wettbewerbe; Hans-Uli Matter, der in einer kritischen Phase zu uns stieß und der die

Entwicklung neuer Lehrinstrumente vorantrieb; Christian Müller, der am Projekt Eileen Gray arbeitete; Sharon Refvem aus Kalifornien, die in der Lehre, im Programmieren und in der Organisation Hervorragendes leistete; Urs Roth für seinen Beitrag und seine Anregungen; Antonino Saggio aus Italien, mit dem ich bereits an der Carnegie Mellon University zusammenarbeitete und der mit Begeisterung Beiträge im Unterricht und für dieses Buch beisteuerte sowie bei der Ausbildung der ersten Nachdiplomstudentinnen und Studenten eine wichtige Rolle spielte; Eric van der Mark, der sich zum Visualisierungsspezialisten entwickelte und von dem die meisten Darstellungen dieses Buches stammen; Claude Vezin aus Frankreich, der durch seine große Geduld den ersten Generationen von CAAD Studentinnen und Studenten das Lernen erleichterte; Daniel von Lucius aus Deutschland, der einen neuen Kurs entscheidend mitentwickelte und unterrichtete; Florian Wenz aus Deutschland, der neue Ideen für die virtuelle Realität einbrachte und maßgeblich beim Aufbau des Architectural Space Laboratory mitarbeitete.

Dank auch den Kolleginnen und Kollegen Professoren an der Architekturabteilung der ETH Zürich: Mario Campi, mit dem zusammen wir unsere neu entwickelten Methoden im Entwurf und in der Praxis testen konnten; André Corboz, mit dem die Arbeit am Schweizer Städteatlas begann; Alexander Henz, der als erster seinen Studentinnen und Studenten die Nutzung von CAAD im Diplom erlaubte; Hans Hugli, dessen Mitarbeiter ein computerbasiertes System zum Erlernen der Statik initiierten; Peter Jenny, der immer die wichtigen Fragen stellte und Unterstützung im richtigen Moment gab; Hans Kollhoff, mit dem ein reger und praktischer Austausch über die Nutzung von CAAD im Entwurf besteht; Herbert Kramel, der meine Arbeit an der ETH von Anfang an mit kritischer Unterstützung begleitete;

Paul Meyer, mit dem ich in einem Forschungsprojekt für integrierte Planung arbeite; Georg Mörsch, mit dem ein Projekt zwischen Denkmalpflege und CAAD über Eileen Gray begann; Werner Oechslin, mit dem die Ausstellung Lineamenta-CAAD zustande kam und der ein unerschöpfliches Wissen über die Verbindungen zwischen Instrument und Architektur besitzt, das sich in seinem Beitrag andeutet; Franz Oswald, mit dem wir ein interdisziplinäres Forschungsprojekt innerhalb der ETH Zürich starteten; Fabio Reinhart, der den Mut hatte, ein von uns empfohlenes System einzusetzen; Heinz Ronner, der wichtige methodische Anregungen machte; Klaus Daniels und Bruno Keller, die mir neue Ideen für die Visualisierung von Aspekten in Haustechnik und Bauphysik vermittelten; Benedikt Huber, Ernst Studer, Arthur Rüegg, Dolf Schnebli, Helmut Spieker und Flora Ruchat-Roncati, deren interessierte und gezielte Fragen mir oft viel Kopfzerbrechen bereiteten und Denkanstöße gaben.

Dank auch an Hans von Gunten für den Beistand im Aufbau der Lehre und an Ralf Hütter für die Hilfe beim Zustandekommen von Forschungsvorhaben; Armin Grün und Hans-Ruedi Schalcher für ihre Initiative in zwei Forschungsprojekten; an meine Kollegen an der

EPF Lausanne, Boi Faltings und Ian Smith, mit denen mein erstes großes Forschungsprojekt an der ETH begann; und an Ulrich Flemming, Volker Hartkopf und Akram Midani an der Carnegie Mellon University. Das Buch wäre ohne meinen Forschungsaufenthalt an der Technischen Universität Dänemarks in Lyngby nicht zustande gekommen. Dort gebührt besonderer Dank den Kollegen Tom Andersen, Egil Borchersen, Per Galle und Niels Carlsen, die mich mit vielen Diskussionen und Anregungen begleiteten. Ohne die unermüdlichen Nachteilsätze von Eric van der Mark, Daniel von Lucius und Florian Wenz, die kreative Aufrechterhaltung der Computerinfrastruktur durch Werner Riniker und die administrative Unterstützung durch Sibylla Spycher wäre dieses Buch nicht rechtzeitig fertiggestellt worden. Ganz besondere Anerkennung gebührt auch den vielen hundert Studentinnen und Studenten sowie den Diplomandinnen und Diplomanden, die uns in den ersten vier Jahren der Professur für Architektur und CAAD mit ihren hervorragenden Arbeiten immer wieder Mut und Freude zum Weiterentwickeln und Verbessern der Theorie und der neuen Instrumente gaben. Ihre Arbeit ist durch viele Beispiele in diesem Buch vertreten und demonstriert den Gang der Entwicklung des CAAD.

Architectura

CAAD und Geschichte - computus et historia

Werner Oechslin

Nehmen wir das Beispiel vom Asphalt. Der Duden gibt immer noch die Schreibweise mit 'ph' und signalisiert damit, daß der Begriff aus dem Griechischen stammt. Doch ob man nun die Asphaltwüste in unserer modernen Welt beklagt und jährlich die Größe zerstörten Kulturlandes aufrechnet, oder ob man die Vorzüge unserer autogerechten Verkehrsstränge lobt, jedermann wird Asphalt als spezifisch zu unserer eigenen industrialisierten, 'modernen' Welt rechnen und darauf schwören, es handle sich um etwas Neues, etwas, das unsere eigene Welt von der vergangenen, historischen, überholten unterscheidet. Daß Jahrhunderte zuvor - und Generationen vor den ersten asphaltierten Straßen - ein beträchtliches Wissen über Asphalt bestand und durchaus zu differenzierten Beobachtungen führen konnte, klingt wie eine Botschaft von einem andern Stern. Und doch trifft gerade dies zu, finden sich 'Asphalt' und 'Bitumen' in jedem enzyklopädischen Werk verzeichnet und ließe sich eine breitangelegte Geschichte dazu rekonstruieren. Völlig entlegen präsentiert so beispielsweise der Orientalist Johann David Michaelis (1717-1791) seine Asphaltkenntnisse in einer Untersuchung zur Natur und zum Ursprung des Toten Meeres ('De Natura et Origine Maris Mortui'), die er am 16. August 1760 vor der Göttinger wissenschaftlichen Gesellschaft vorträgt und in der wir - mit Verweis auf noch viel ältere Literatur - so unterschiedliche Dinge erfahren wie Vorkommen und Brennbarkeit, die dabei beobachtete ätzende Rauchentwicklung und auch die Verwendung in der Medizin. (Zuvor hatte natürlich die Encyclopédie berichtet, daß die 'erste' Auffindung einer Asphaltader in Europa - samt nachfolgender Nutzanwendung im Schiffsbau - im Val de Travers durch Monsieur de la Sablonniere 1740 geschehen sei!). Die Publikation der

gesammelten Göttinger Vorträge Michaelis' schmückte auf dem Titelblatt ein Emblem mit der Überschrift 'fecundat et ornat'. Niemand mag grundsätzlich an der 'befruchtenden' Wirkung der wissenschaftlichen Anstrengungen des Herrn Michaelis zweifeln. Niemand würde dies aber so auffassen, daß diese nutzbringende Funktion bis heute gleichsam kontinuierlich anhalten würde. Denn was Michaelis in seiner Zeit vorwärtsdenkend beibrachte, ist längst überholt und abgestreift, auch wenn uns beispielsweise die grundsätzliche Einsicht in eine industrielle wie medizinische Nutzung derselben organischen Substanzen natürlich weiter interessiert. Doch auch hier bleibt unter dem Strich nichts übrig, was nahelegen würde, beim Gedanken an Asphalt ausgerechnet die Geschichte zu bemühen.

Hätte man zu Beginn unseres Jahrhunderts nach dem Verständnis des Begriffes 'Computer' - oder seines lateinischen Verwandten 'computatio' - gefragt, wäre man allenfalls (mit Hilfe allgemeinbildender Lexika) zum juristischen Fachbegriff der Berechnung von Verwandtschaftsgraden oder zum alten 'Computus paschalis', der Berechnung des Osterdatums gelangt. Und man dürfte heute noch einige Zeitgenossen - unter der kleinen Zahl von Mittelalterspezialisten - finden, bei denen 'computer' die Assoziation zu Beda Venerabilis - und sein zu Beginn der Neuzeit wiederholt aufgelegtes 'De Computo, vel loquela per gestum digitum' auslöst, was nichts anderes ist als eine Anleitung, wie mit Fingerstellungen Zahlen und Buchstaben symbolisch dargestellt werden können. Bis dahin bleibt wohl alles - aus der Sicht des CAD-Adepten - exotische Geschichte, ohne einseharen Zusammenhang. Fügt man allerdings an, daß kein Geringerer als Leibniz bei solchen Modellen ansetzt, dann mögen sich die Augen auch des Nichthistorikers erstmals

weiten. Denn Leibniz ist nun eben einmal gemäß der üblichen historischen Vulgärauffassung (der selbst der gute Pascal meist zum Opfer fällt) der 'Erfinder der ersten Rechenmaschine'. Leibniz hat in einer Zeit (der unsrigen), in der - wiederum gemäß vulgärer Vereinfachung, die allerdings auch bei den Spitzen und Verantwortlichen des heutigen Wissenschaftsbetriebs nicht Halt macht - die Naturwissenschaften nach vorne, die Geisteswissenschaften hingegen nach rückwärts gewandt sind, einen 'besseren Klang' als manches, was so aus der Geschichte daherkommt. Natürlich vergessen bei solcher Voreingenommenheit die meisten, daß Leibniz in erster Linie auch Historiker war. Sein sprachhistorisches Interesse brachte ihn zu solch merkwürdigen - mit Bedas 'computus' vergleichbaren - Phänomenen wie dem von den Zisterziensern verwendeten Handzeichen-Alphabet. In den 'Collectanea etymologica' des 'Godofr. Guilielmi Leibnitii' wurden 1717, kurz nach Leibniz' Tod, solche auf nicht mehr als ca.140 Zeichen beschränkte Kataloge von Handzeichen gleich in drei Versionen, einer lateinischen, deutschen und spanischen publiziert. (Das lautete etwa so: "Domus, facere quasi tignum cum digitis manuum", oder: "Sperestu beyde hände thohope dackes wyse, dat is ein Hues.") Im Index der 'Collectanea' ist dieses Kapitel präzise und knapp betitelt mit "Signa, per quae aliquis mentem suam alteri sine loquela indicare potest"; oder frei übersetzt: Zeichen(sprache), durch die jemand einen Gedanken ohne Rede einem anderen mitteilen kann. Leibniz hat schon in seiner Frühschrift zur 'ars combinatoria' (1666) sein Interesse an der Rückführung komplexer Gegebenheiten auf einfache Begriffe bekundet, mittels derer Neues zu erreichen wäre. Er schloß an die alte 'ars inveniendi' an, sah in der Synthese der 'art d'inventer' und der 'méthode de la certitude' die Aus-

sicht auf eine 'science générale' gegeben und suchte in diesem Sinne zeitlebens nach einer entsprechenden - teils 'mathematischen' - Universalsprache (gemäß der Annahme: "Sed nihil est quod numerum non patiat. Itaque numerus quasi figura quaedam Metaphysica est.. qua rerum potentiae explorantur.") Leibniz hat bei solchen Untersuchungen an die systematischen Leistungen von Geistesgrößen angeknüpft, die bis heute - in der Tradition eines positivistischen Wissenschaftsverständnisses - mit größter Skepsis bedacht sind: an Raimund Lull und Athanasius Kircher.

So gleiten wir also schon wieder ab vom gerade noch hoffähigen Leibniz zu Positionen, die im Dunkel der Geschichte - wie viele denken, völlig zurecht - verschwinden. Dies geschieht wie immer nach dem Prinzip, gemäß dem die durchlöchernten Socken Charlie Chaplins einen vom Auktionator festzulegenden Preis garantieren, wogegen diejenigen des Schreibenden und wohl auch des Lesenden in doppeltem Sinne nur Nasenrümpfen hervorbringen. Das Interesse an - historischen - Dingen entwickelt sich auf verschlungenen, meist nicht besonders einsichtigen Pfaden. Und damit läßt sich 'in wissenschaftlicher Hinsicht' nur wenig anfangen. Es bedarf zumindest hier, eines besser begründeten Zugangs, um das zu vermeiden, was üblicherweise in solchen Situationen (nach 'postmodernem' Muster) einsetzt: die zufällige oder gar beliebige Berücksichtigung des einen oder anderen, mit Sicherheit aber andern Zwecken (der Legitimation oder auch nur Kuriosität) zugeführten Bezuges zur Geschichte.

So wie die Auseinandersetzung des Neuen mit dem Alten - auch bei CAD oder CAAD - meistens verläuft oder verliert, mag nur wenig befriedigen. Was sagt es schon aus, wenn der Rückblick aus unserer heutigen

Computerwelt bei der Rechenmaschine von Leibniz, oder bei derjenigen Pascals oder Giovanni Polenis halt macht. Mehr als ein billiger Autoritätsbeweis ("schon der berühmte Leibniz hat eine Rechenmaschine erfunden!") ist das - vorgängig genaueren Hinsehens, womit wir eben ansetzen! - nicht. Damit ist vielleicht ein Stück Kuriosität befriedigt oder allenfalls das, was gemäß einem alten Gemeinplatz Ciceros als Aufgabe der Geschichte stets vermerkt wurde: 'nuntia vetustatis', Kunderin des - ehrwürdigen - Alters. Anders und weitaus schwieriger wird es, wenn man der anderen ciceronianischen Anforderung an die Geschichte nachgeht, die mit der 'magistra vitae', der Lehrmeisterin des Lebens apostrophiert ist.

Davor hat heute selbst der Historiker meistens Angst. Er flüchtet sich in die Geschichte und legitimiert sich unter seinesgleichen (das tut der Wissenschaftler auch). Oder aber er geht an die Front und verlegt sich auf das, was mit dem neudeutschen Wort 'Akzeptanz' treffend bezeichnet und auch gleich eingeschränkt wird. Bei ersteren ist kaum was zu holen für unsere Fragestellung; bei letzteren bewegt sich das Potential, das Neue in dem nicht bereits Akzeptierten, Vermittelten und Vermittelbaren auszuloten, gegen Null. Die Ausgangslage favorisiert den mühsameren dritten Weg also kaum. Und dies darf - jetzt mit besonderem Blick auf CAAD - noch weiter illustriert werden.

Man kann nicht behaupten, daß seit Aufkommen von CAAD der intelligente Gebrauch des neuen Mediums - diese Sprachregelung ist wohl hinlänglich verständlich, seitdem 'intelligentes Bauen' als 'andere Baukultur' angepriesen wird - bei Architekten überwiegt. Zumindest zwei Faktoren überlagern einen solchen qualifizierteren Zugang. Zum einen ist CAAD längst Prestigeangelegenheit. Ein renommiertes Büro

kann es sich schon längst nicht mehr leisten, dem Bauherrn ein Projekt in konventionellen Darstellungsformen vorzustellen. Eine CAAD-Simulation muß her. Sie übernimmt in diesem Sinne - und dies ist der zweite Faktor - die klassische Funktion des Modells, von dem man schon immer wußte, daß es gerade dazu taugt, dem Nichtfachmann die Idee sinnhafter und suggestiver, mit mehr Überzeugungs- und Simulationskraft vorzutragen. Insofern bringt CAAD natürlich (noch) nichts Neues, sondern vollbringt in zeitgemäß modischer Art das, was zuvor auf andere Weise geleistet wurde. Da nützt es nichts, daß schon Alberti um 1450 monierte, im Modell den 'Geist der Erfinders, nicht aber die Hand des Verfertigers' zu bewundern! Modelle sind verführerisch und ästhetisch reizvoll. Was der 'renderer', der spezialisierte Darsteller, in der ersten Jahrhunderthälfte mit hoher Virtuosität leistete, das soll jetzt, wenn es denn nach der Vorstellung so manch eines Architekturbüros ginge, der CAAD-Spezialist garantieren: Er ist Darsteller und Verkäufer, dem Wunschdenken des Architekten entsprechend subsidiär dienend!

Man braucht nicht weit zu blicken, um dieses Phänomen am repräsentativen Ort illustrieren zu können. Die 'flotten Darsteller', um einmal mehr an den Begriff Adolf Loos' zu erinnern, sind längst wieder unter uns. Ein Meister dieses Faches, Peter Eisenman, hat sich schon früh damit gebrüstet, Projekt, Ausstellungstafel und Layout für die Publikation in einem Arbeitsgang zu bewältigen, wogegen nur ein Moralist grundsätzlich etwas einzuwenden hat. Jetzt sucht er folgerichtig, seine im Entwurf zu wählende architektonische Form dem - spezifischen - Computerbild (und nicht umgekehrt) anzunähern und kann sich so sicher sein, die Sensibilität seiner eigenen Zeit, den 'Zeitgeist' mitten ins Herz zu treffen. Letztes Resultat

dieser kulturbewußten Architektur: der Vorschlag einer Computergraphik als Hochhaus in Berlin, die prompt auch schon mal als 'ein Monument des 21. Jahrhunderts' und als 'epochales Manifest' apostrophiert worden ist, im übrigen aber - bis zur nächsten PR-Aktion - längst wieder vergessen ist. Eisenmans 'Max-Reinhardt-Haus', so heißt die Computer-Graphik, ist zumindest provokativ und versucht, spezifische CAAD-Formen auch in ihrer unverkennbaren ästhetischen Wirkung aufzunehmen. Doch es bleibt wesentlich dabei, daß auf dem Weg des Bildes eine andere als die gewohnte Architektur suggeriert wird.

Bei den meisten Kollegen zeigt sich dieses Vorgehen schon gar nicht erst durch eine spezifische Andersartigkeit begründet; der CAAD-Effekt reduziert sich dort meist auf die bildliche Umsetzung (und Aufmöbelung) einer durchaus konventionellen architektonischen Erfindung. Der Preisträger des jüngsten Berliner Wettbewerbs für den Spreebogen, Axel Schultes, hat sein Projekt unter anderm in einer großen Schautafel einer Computer-Simulation dargestellt. Das Bild hat Anklang gefunden, sich offensichtlich - nicht nur bei den Juroren - durchgesetzt und geht nun wie ein Fanal durch die Gazetten, obwohl - oder eben gerade weil - es die bildimmanenten (nicht die architektonischen) Vorzüge nach außen kehrt. Die abstrakt summarisch gegebenen Bauten Berlins erscheinen rot, Spree und Spreehafen violett, die Schatten grün. Das Projekt im engeren Sinne hebt sich ebenso abstrakt reduziert durch hellere Töne ab. Fazit: Das Bild wird zum Faszinosum. Genauer genommen: die ihr eigene neue - und noch nicht alltägliche - Darstellung in geometrisch und farblicher Verfremdung!

Dies alles formt ein Kapitel einer künftigen Kulturgeschichte des CAAD - für sich genommen durchaus interessant. Es beschreibt eine im übrigen längst

ersichtliche ästhetische Sensibilität, die im Kreis der 'Dekonstruktivisten' - und ihrer Ahnen bei de Stijl und anderswo - längst vorbereitet war. Kein Zweifel, diese Phänomene verdienen unser Interesse. Allein, um wieder auf unser eigentliches Thema zurückzukommen, sie dürften wohl schwerlich als Zielpunkt jetziger und künftiger - systematischer - Bemühungen im Zusammenhang mit CAAD ausgegeben werden. Denn noch bleibt trotz einiger Unkenrufe der Anspruch erhalten, daß der Architekt mehr anstrebt, als bloß der 'flotte Darsteller' vom Dienst zu sein. Und das ist recht so.

Noch ein Aspekt solcher 'Eigendynamik' verdient unser Augenmerk. Er betrifft am andern Ende des Spektrums möglicher Wirkungsgrade den in der Tat höheren und hohen Anspruch, mit CAAD letztlich - gemäß einer jetzt noch bewußt 'floppigen' Formulierung - andersartige, neue Architektur schaffen zu können. Just auf solch unverbindlich-unpräzisen Pfaden verlief weltweit eine öffentlich ausgetragene Diskussion über Sinn und Unsinn des Computers, als ob man - fernab von Geschichte und ihren Erfahrungen - zum ersten Mal einer solchen Problematik ausgeliefert worden wäre. Sowie die Geschichte von CAD, wo sie denn doch einmal in die Lücke getreten ist, meist eine Geschichte der Rechenmaschine - etwas eleganter: 'une histoire des ordinateurs' - geblieben ist, ohne wesentlich darüber hinaus (nach geistigen Beweggründen, Modellen u.a.m.) zu fragen, so platt verläuft meist die Diskussion über die 'philosophischen' Implikationen. Da ist uns längst eine 'neue Denkweise' suggeriert worden, die vom Vergleich mit unserem 'alten' Denken meist eher ablenkt. Andererseits hat man den Computer nicht gerade zimperlich 'geradezu eine Ausstülpung unseres menschlichen Gehirns' genannt. Und von der Papertschen Definiti-

on des Computers als eines Dinges, 'mit dem man denkt', ist auch schon mal - schwarz auf weiß nachzulesen - gefolgert worden, das Gehirn sei doch 'hoffentlich ebenfalls ein Ding, mit dem man denkt'. Die Argumente erinnern öfters an klassische Diskussionen, wie jene zur Zeit der Einführung der Maschinen! Ersetzt sie den Menschen? Übernimmt sie Entscheidungen? Das Fließband kam und - in Europa verspätet - ging. Und schon wieder gerät der Mensch in Zweifel, ob mit dem neuen Instrument seine Gehirntätigkeit Konkurrenz erfährt oder gar ins Hintertreffen gerät... Es wird wohl der Phantasie des cleveren Verlegers zuzuschreiben sein, wenn der Disput zwischen zwei so prominenten Wissenschaftlern wie Klaus Haefner und Joseph Weizenbaum auf den Titel 'Sind Computer die besseren Menschen?' verkürzt wird. Damit gewinnt man Käufer und kann sogar ein Fernsehpublikum anziehen. Doch, was in solchen Disputen tatsächlich zum besten gegeben wird, findet denn doch meist im Vorhof wirklicher geistiger Anstrengung statt: 'small-talk' über Lebensgewohnheiten, ob ich beispielsweise meine Briefe auf Grund von Training des visuellen und akustischen Gedächtnisses (sic!) diktiere oder weiterhin von Hand schreibe. Natürlich soll man 'Computer-Ängste' nicht völlig übersehen und kann man 'philosophisch-ethische' Zweifel ernst nehmen. Aber man mußte auch bedenken, wieviele dieser Ängste gerade durch eine unqualifizierte Diskussion - zur 'Computerisierung des Alltags' beispielsweise - geschürt werden. Nichts von all den Horrorvisionen, die mit dieser Pauschalisierung beschrieben wurden, ist Realität geworden. Allein, der Disput - zwischen Haefner und Weizenbaum, um dieses populäre Beispiel vergleichsweise anzuführen - verläuft meistens auf der Ebene solcher Verallgemeinerung. Da geht es darum,

ob Computer Informationen 'produzieren' und ob man daraus nicht allzu schnell folgern könnte, sie würden auch Wissen erzeugen oder sogar Bildung hervorbringen (der klassische Stoff eines Science-Fiction-Romans!). Da wird die Möglichkeit des 'bewußten Rechners' suggeriert, das tönt wie 'glückliche Kühe', erinnert an Formulierungen, die Poeten beherrschen und für die die klassische Rhetorik präzise Begriffe führt! Und aus der Beobachtung, Mozart hätte mit einer Sonate eine neue Tonfolge gefunden, die nun gleichsam zu den bis dahin bekannten Bausteinen von musikalischen Kompositionen hinzugelegt werden könne und so - so die umwerfende Logik - Musik völlig berechenbar machen würde. Zweifelsohne, im Kampf der Titanen um den 'besseren Menschen' sympathisiert der Schreibende mit Weizenbaum. Um zu erfahren, was denn bei einem Denkvorgang, den zu simulieren der Computer ja vorgibt, als 'neu' zu bewerten sei, möge man dringlichst die veralteten spätmittelalterlichen Vorstellungen - zum 'Syllogismus' in Erinnerung rufen: Sie sind zumindest nach Maßgabe des erwähnten Disputs völlig ausreichend! So würde man etwa erfahren, daß eine Figur natürlich jedesmal - auf dem Reißbrett oder dem Bildschirm - neu entsteht, wenn ich die dafür notwendigen geometrischen Teile in letztlich nie endender Variation zusammensetze. ('Figura est, quae ab aliquo, vel aliquibus Terminis comprehenditur.') Der einfache Syllogismus schafft - 'more geometrico' - etwas Neues, selbstverständlich. Nur muß ich deshalb noch lange nicht in die Euphorie oder den Pessimismus deterministischer Teufelskreise geraten. Man kann auch kaum die Begeisterung des Herrn Haefner teilen, der darauf wartet, daß der Mensch 'aufgrund gigantischer Computerberechnungen' eines Tages wissen wird, wievie-

le Primzahlen es gibt. Es scheint vergessen, daß zu den intelligentesten Leistungen jene gehören, die Begrenzungen aufzeigten oder gar Unbewiesenes als Unbeweisbares enträtselten. Giulio Camillo hat schon im 16. Jahrhundert den Spieß umgedreht und sich den verblüfften Fürsten vorgestellt, dem er anbieten würde, mit bloß 22 Elementen alles Sichtbare und Unsichtbare darstellen zu können. Es gehörte damals zu den anerkannten Grundsätzen, daß eine kleinere Zahl von Regeln und Prinzipien eine um so universellere Geltung beanspruchen dürfte und deshalb vorzuziehen wäre. Man hatte Alphabet und Zahlen in ihrer Funktion durchschaut. Man wußte - um beispielsweise den Lehrmeister Palladios, G. G. Trissino zu erwähnen - bezüglich Sprache, Alphabet und Schriftbild zu unterscheiden zwischen 'Principi primi', 'elementi', 'note' und 'lettere' und es spricht vieles dafür, daß dieser 'analytische' Zugang zur Sprache auch auf einen ähnlich qualifizierten Umgang mit Architektur Auswirkungen hatte...

'Mens agit at molem'! Natürlich gibt es - neben dem Wust boomartig sich breitmachender Trivialliteratur zum Thema Computer - auch sehr viele ernst zu nehmende Äusserungen zu Denkform und künstlicher Intelligenz. Fragen wir beispielsweise den zuständigen Neurologen. Er trifft die klare Unterscheidung, daß der Computer 'in bezug auf Rechengeschwindigkeit, Speicherkapazität, Datenzugriff' dem menschlichen Gehirn überlegen sei, daß ihm jedoch 'charakteristische Eigenschaften biologischer Intelligenz' - 'bewußtes Wahrnehmen, Ich-Begriff, Einheit des Bewusstseins, Selbstorganisation' - fehlen. Letztere Eigenschaften dürfen wir ohne allzu große Risiken für all jene menschlichen Leistungen in unterschiedlicher Intensität und Ausformung

beziehen, die sich historisch vollzogen haben und sich weiter vollziehen. Was für ein Glück ist es doch, daß neben den Extremen euphorischer Vorstellungen zu mathematischen Automatismen, in die der menschliche Geist im Aufbruch zum neuen Zeitalter des Computers eintaucht, und neben denjenigen hypersubjektiven, antirational fundamentalistischen Zuschnitts auch noch Mittelpositionen zu finden sind, die in der 'ratio' ein Instrument und im 'more geometrico' - weiterhin - Methode und nicht Selbstzweck sehen. Wie war das doch so schön angelegt in Herders 'Ideen zur Philosophie der Geschichte der Menschheit': "Der Mensch ist zu feinem Sinnen, zur Kunst und zur Sprache organisirt." Das war keine mathematische, sondern eine kulturgeschichtliche Aussage, der die Mathematik jedoch sehr wohl wichtiger Bestandteil war.

Genau darauf kommt es an! Die 'Computer-Welt' abgelöst, isoliert zu betrachten führt letztlich nicht weiter. Beide Extreme, die einseitig künstlerisch-ästhetische Aura, der sich der Computer - letztlich zum Beweis seiner kulturellen Bedeutung - längst bedient hat, und die abgelöste, dem Traum einer autonomen, quasi-mathematischen Generierung blind gehorchende Vorstellung einer 'neuen Denkweise' bedürfen der Einbettung in kulturelle Realitäten - um so mehr, als sie ihre Anwendung in solchen Bereichen wie der Architektur finden sollen. Was liegt da näher als in der Geschichte Entsprechungen, Analogien und Widersprüche zu vermuten, auf Grund deren diese Situierung verständlicher wird und andererseits neue, bestätigende oder abweichende Horizonte gezielter angepeilt werden können. Dies ist wohl die Minimalforderung des Historikers, die er einzubringen aufgefordert ist und die ihm sein 'magistra vitae' auf ein vernünftiges Maß

beschränkt einlöst: Es soll die Qualität des Neuen im Vergleich mit dem Alten deutlich gemacht werden. Dies würde den Rest unseres Fortschrittsglaubens - was anderes insinuiert denn die 'magistra' - am ehesten als glaubwürdig am Leben erhalten. Und bezogen auf den obigen Gegensatz: Es wird gerade darauf ankommen, daß man nicht wieder voreilig den 'gefühlsorientierten' Teil CAAD-produzierter, suggestiver Bilder von einer objektiv-wissenschaftlichen Grundlage abtrennt, sondern daß man umgekehrt versucht, den Zusammenhang von Wissenschaft und Kunst in den Griff zu kriegen. Dies ist zugegebenermaßen ein hohes Ziel, aber auch nicht so völlig außer Reichweite. Es mag manchmal scheinen, daß eine frühere Zeit ein breiteres Bewußtsein bezüglich unterschiedlicher geistiger Funktionen besessen hätte, als dies unsere 'moderne' Opposition von Gefühl und Verstand suggeriert. Kreativität, Phantasie, Intellekt verstanden sich nicht als Gegensatz, sondern als unterschiedliche, zu koordinierende Funktionen ein und derselben Geistestätigkeit. Heute glaubt jede Mutter zu wissen, daß ihrem Kind mit dem Lernen gleich die Phantasie und Kreativität ausgetrieben würde.

Der verpönte Robert Fludd hat in seinem Schema zu den Funktionen des menschlichen Gehirns unseren modernen Gehirnsanalysen nichts Adäquates entgegenzusetzen, doch wußte er 'Anima sensitiva', 'Anima imaginativa' und 'Anima cognitiva' und auch noch die 'Memoria' so miteinander zu verknüpfen, daß daraus ein hinlänglich umfassendes Modell unserer unterschiedlichen, vernetzten geistigen Befähigungen zustande kam. Der Neurologe hat dafür - so will es scheinen - mehr Verständnis als der Computerefachmann.

Aus diesen ausufernden Überlegungen darf man vorerst theseartig - immer mit Blick auf Architektur und CAAD - einige Postulate formulieren:

1. Wenn es zutrifft, daß Architektur stets kulturell gesellschaftlich eingebunden war und sie diese Ausrichtung nicht nur passiv erlitten, sondern aktiv programmiert hat, wenn es deshalb plausibel ist, daß dies wohl auch in Zukunft so sein werde, so gibt es keinen Grund, daß sie sich nun gleichsam auf einen rein-mathematischen Standort zurückziehe. Das bedeutet in keiner Weise, daß sie auf das verzichten soll, was als 'wissenschaftliche Erörterung' stets notwendig und erwünscht ist. Aber sie sollte sich mit Vorteil gerade dieser schon stets gegebenen Situation bewußt sein. Um eines von vielen Beispielen zu geben: J. H. Lambert hat zur Einführung der Unterscheidung historischer und wissenschaftlicher Erkenntnis - wen überrascht's - den 'Erfinder der Geometrie' zitiert, der gerade so gut der entwerfende Architekt (im Bild des Demiurgen) sein könnte. Dieser "nahm etwann drey Linien, und suchte sie in Form eines Triangels zuammen zu legen. Der Versuch gelingt, die Linien schließen sich, und er bemerkt, daß nun die Winkel schon da sind, und daß, wenn er einen derselben, oder zween oder alle drey ändern will, so gleich auch an der Länge der Seiten etwas geändert werden müsse. Die Sache macht ihn aufmerksam. Er nimmt andere Linien, und findet, daß es Fälle gibt, wo sie sich nicht schließen wollen, und daß er eine der kürzern verlängern müsse etc. Man sieht leicht, daß ein kleiner Versuch von dieser Art zu fernern Betrachtungen führt, und daß man unvermerkt verleitet wird, Gründe zu suchen." Natürlich hat Lambert in seiner Zeit - mit Blick auf Galilei oder Newton oder auch Christopher Wren - von einer Ver-

wandlung von 'bloß historischer Erkenntnis in eine wissenschaftliche' gesprochen - und mittelbar den möglichen Zusammenhang von 'Erfahrungsbegriffen' und systematischen Erkenntnissen bekräftigt. Lambert bietet nur ein kleines Fragment aus jener umfassenden Geschichte, der die Mathematisierung des Denkens (der Architektur verständlicherweise nahe) oder auch bloß die mathematische Metapher eigen ist. "Le nombre est la barrière évidente entre la brute et nous; dans l'ordre immatériel, comme dans l'ordre physique", läßt J. de Maistre in seinen 'Soirées de Saint-Pétersbourg' vernehmen. Und Vitruv sagt nicht nur, der Architekt müsse Kenntnisse in Mathematik erwerben, sein Interpret Daniele Barbaro hat schon längst (1556) das Menschenspezifische - in der Abgrenzung zu den 'bruta' - über die Auslegung von 'ratiocinatio' auf die spezifische Beschaffenheit einer Theorie der Architektur hingelese. Die Zusammenhänge sind zumindest kulturgeschichtlich erwiesen - und liegen für jede neue Betrachtung der Dinge brach.

2. Damit sind wir längst bei der Architektur selbst. An solchen grundsätzlichen Vorstellungen und Modellen zur menschlichen Vernunftstätigkeit, zum Vergleich von 'historischer' und wissenschaftlich-systematischer Erkenntnis ist gerade die Architekturgeschichte - genauer: die Geschichte ihres Theorieverständnisses - voll. Und es ist wichtig, sich daran zu erinnern, daß die Architektur, seitdem sie sich im 15./16. Jahrhundert als Disziplin mit wissenschaftlichem Anspruch formiert hat, selbst einen solchen 'philosophischen' und 'wissenschaftlichen' Rahmen gebildet hat. So besehen, bietet die Geschichte einen Fundus nicht bloß an Fakten, sondern eben insbesondere auch an Denkmodellen, Theorien und Methoden und - um näher an der CAAD-Praxis zu argumentieren - an For-

men von Zeichensystemen, Matrixen, Klassifikationsmodellen, Darstellungsmethoden u.a.m.

Zu diesem Rahmen gehörte auch schon immer die besondere Fragestellung nach Wissenschaftlichkeit und Kohärenz - gerade bezüglich der zeichnerischen Erfassung des architektonischen Gegenstandes und bezüglich eines systematischen Entwurfsvorganges. Denn dieser war schließlich schon immer das A und O der Architektur, seitdem sie der bloßen Erfahrung des Baumeisters enthoben wurde. Was ein 'Bauhüttengeheimnis' war - um es arg zu vereinfachen: vertraulich gehandhabte Modalitäten im Umgang mit Zirkel und Lineal -, wird zur Methode, und darauf ließ sich aufbauen. Natürlich wurde wie immer vieles, was auf systematischem Wege erarbeitet wurde, zur Konvention. Die 'Regola' des Vignola, ein erstaunliches Konstrukt in der Verbindung von Maßangaben und Proportionen und insgesamt deren Vergleichbarkeit, konnte nun einmal schnell zur Rezeptur verkommen, so wie heute natürlich viele ihr Auto fahren, ohne eine Ahnung vom Funktionieren eines Motors zu besitzen. Der große Gedankenaufwand, der bei der Entstehung solcher Modelle Pate gestanden hat, wurde natürlich - zumindest in der Praxis - meist vergessen. Das macht ihn nicht ungeschehen. Die Tradition läßt sich aufzeigen und notfalls rekonstruieren. Man denke an ein so selbstverständliches Problem wie die Zerlegung einer Architekturdarstellung in Grundriß/Aufriß/Schnitt (was zwischen 1500 und 1560 zu erheblichen Diskussionen geführt hat, bevor es einigermaßen verbindlich kodifiziert war). Noch in diesem Jahrhundert hat im Vergleich 'plan libre/façade libre' bei Le Corbusier und 'Raumplan' bei Adolf Loos, ersteres den Sieg davongetragen: das analytische Modell, die Umle-

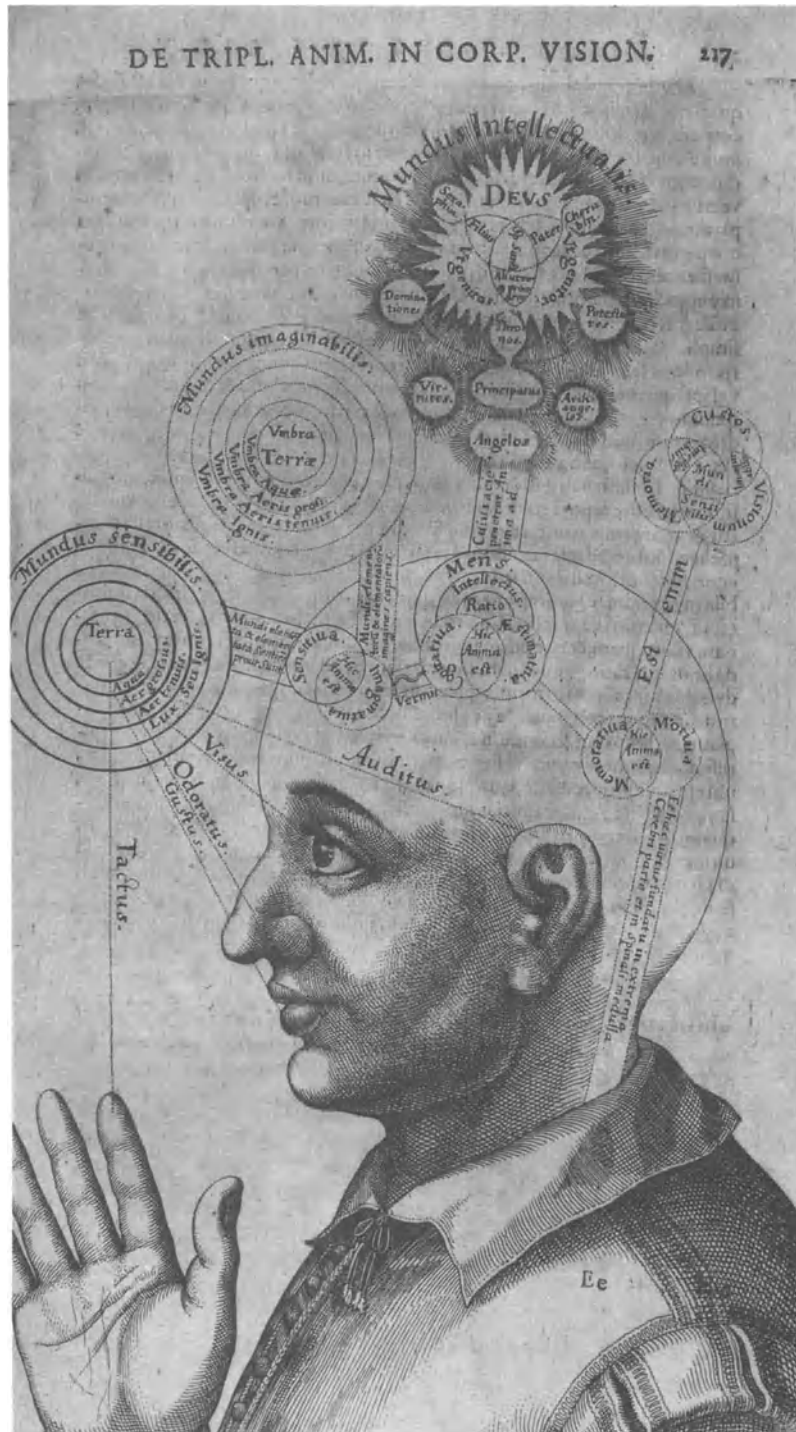
gung des dreidimensionalen Objekts in die Flächen hat - noch einmal - den Angriff einer direkten zeichnerischen Kodifizierung des Dreidimensionalen abgewehrt, ist über die Axonometrie vorerst nicht hinausgekommen. Jetzt mit CAAD steht die Frage wieder neu zur Disposition.

Geschichte! Es gibt kaum ein Problem, das sich nicht auf diese Weise einbinden ließe in eine lange Folge von Fragestellungen und Positionen. Und natürlich kann es nicht darum gehen, dies aus bloßer Kuriosität zu tun. Es rechtfertigt sich vielmehr in dem Maße, wie Orientierungshilfe zu erwarten ist. Und wie kann dies geschehen? Geschichte hat den Vorzug, wenn man Augen hat zu sehen, daß sie stets selektive Prozesse gekannt hat. Der Vergleich theoretischer Modelle mit der Wirklichkeit hat schon immer deren Anwendungsradius - selektiv - eingeschränkt. Diesen kleinen Vorteil kann man erhaschen. Ein Beispiel: Es ist weiter kein besonderes Kunststück, mit CAAD auf Grund weniger vorgegebener Parameter zig-tausend Kombinationen beispielsweise für die grundrißliche Anordnung eines Einfamilienhauses zu generieren. Dagegen steht der verrufenste aller 'simplificateurs' unter den Architekturtheoretikern, J. N. L. Durand in äußerst bescheidener Fassung da. Er zeigt die ersten Schritte einer (weiterführenden) Kombinatorik, beschränkt sich jedoch auf wenige 'exempla', heißt: Er demonstriert die Methode - und verliert sich nicht in die Kasuistik 'aller möglichen Lösungen', die wie er weiß auf Grund bestehender Konventionen ohnehin längst eingeschränkt sind. Natürlich kann ich letztere als Bedingungen in mein Programm eingeben. Und genau da liegt der springende Punkt: Woher denn sonst wenn nicht aus der Erfahrung und Geschichte bringe ich solche zusätzlichen Kriterien, samt ihrem relativen Stellenwert ein?

3. Wir sind längst mitten in einer konkreten Argumentation, die eine systematischere Darstellung verdiente. Doch bleibt ein dritter grundsätzlicher - ebenfalls bereits angesprochener, aber hier zu präzisierender - Punkt. Es ist bedeutsam - und leider viel zu wenig bekannt -, daß sich die Architekturtheorie in ganz besonderem Maße immer wieder um jene spezifische Frage gekümmert hat, der sich CAAD ebenfalls in erster Linie zuwendet: der abstrahierenden, modellhaften und antizipierenden Darstellung von Architektur im Mittel geometrischer Zeichnung, zusammengefaßt im (systematischen) Entwurfsvorgang. Alle diese Elemente sind ausführlichst bedacht worden: der Abstraktionsgrad in der 'linearis essentia' (respektive der Kompromiss in der sinnlich erfahrbaren Linie), der Modellcharakter, das 'im Geiste Vorwegnehmen' - was jetzt beispielsweise als 'virtuelle Realität' neu umschrieben wird -, der besondere geometrische und insofern verlässliche Charakter der Architekturzeichnung bzw. die entsprechende Approximation anderer, figürlicher Teile, und natürlich die Zeichnung selbst in all ihren Formen, in ihrer Funktion der Kodifizierung, ihrer klassifikatorischen Möglichkeiten und schließlich die Zusammenfassung all dieser Einzelaspekte und Möglichkeiten in dem Form (neu) schaffenden Prozeß des Entwurfs.

An dieser Stelle müßte die konkrete und detaillierte Argumentation einsetzen, wie sie in der Ausstellung 'Lineamenta - CAAD' auf Grund weniger Beispiele stellvertretend dargelegt wurde und wie sie der Autor in seiner Forschung seit Jahren verfolgt. Damit würden wir jedoch das 'computus ET historia', den Annäherungsversuch zu einem Thema und die Anregung zu einem Vergleich längst verlassen. Dies sei 'computus CUM historia' im Sinne der Absicht des Autors vorbehalten.

Robert Fludd,
Tomus Secundus de Supernaturali, Naturali, Praeternaturali Microcosmi historia,
'Trium Animae Visionis generum in Microcosmo, eorumque regionum, objectorum,
rationumque discernendi luculentissima demonstratio', Oppenheim, 1619, S. 217



Architectura - eine Wissenschaft

Architectura, das lateinische Wort für Architektur, gewinnt im Zusammenhang mit dem Thema CAAD eine neue Bedeutung. Es sind darunter alle jene Schritte zu verstehen, welche die Existenz eines Gebäudes, beginnend bei dem Prozeß des architektonischen Entwerfens, über die Simulation des Gebäudes, zu seiner Fertigstellung bis hin zu seinem physischen Ende begleiten. Die elektronische Spiegelung dieser Schritte in einem neuen Medium wird den Prozeß der Entstehung und das Erscheinungsbild der Architektur verändern und damit eine neues Architekturverständnis eröffnen.

In diesem ersten Kapitel wollen wir uns auf den Entstehungsprozeß von Architektur und dessen Unterstützung durch CAAD konzentrieren. Architektur ist das vieldimensionale Ergebnis eines intellektuellen Prozesses, der in Komplexität und Intensität einmalig ist. Zwar schreibt Rudofsky von Architektur ohne Architekten [Rudofsky 1964], doch entsteht Architektur nie ohne Intention, ohne Willen, ohne geistigen Prozeß. Die Beschreibung dieser Prozesse ist der erste Schwerpunkt dieses Kapitels.

Architektur als Wissenschaft ist ein Thema, das sich durch die Jahrhunderte zieht und das zunehmend an Brisanz und Relevanz gewinnt. Architektur lebt auch von den Fortschritten der Wissenschaft. Zugleich befruchtet die Architektur durch ihre Konstanz über die Zeit und den dynamischen und kreativen Prozeß ihrer Entstehung die Wissenschaft. Dennoch ist Architektur insgesamt als wissenschaftliches Forschungsgebiet nicht mehr oder noch nicht anerkannt. Zwischen Architektur und Wissenschaft scheint ein Spannungsverhältnis zu bestehen. In den USA wird Architektur oft im Bereich der Schönen Künste (Fine Arts) gelehrt. Dementsprechend sehen sich viele Architektinnen und Architekten als Künstler und lassen sich von den tech-

nischen Aspekten der Architektur eher abschrecken. Einen Nobelpreis in Architektur gibt es nicht. Trotzdem: Das Erschaffen von Bauwerken umfasst wichtige Aspekte wissenschaftlicher Forschung.

Ergebnisse wissenschaftlicher Forschung müssen nachvollziehbar, quantifizierbar und wiederholbar sein. Forschung in den technischen Bereichen der Architektur folgt diesen Vorgaben und erfüllt die Kriterien, wie den zahlreichen Publikationen auf diesem Gebiet zu entnehmen ist. Als Beispiele seien nur die Statik, die Haustechnik und die Bauphysik erwähnt. Gebaute Architektur erfüllt diese Bedingungen jedoch ebenfalls: nachvollziehbar, da dokumentiert und geplant, quantifizierbar, da in Substanz und meßbaren Größen vorhanden, und wiederholbar, da in einem dokumentierten und vorausbestimmten Prozeß entstanden. Zusätzlich erfüllt neue Architektur oft das Kriterium der Innovation. Damit weist gebaute Architektur ähnliche Eigenschaften wie die Ergebnisse anderer wissenschaftlicher Forschung auf. Der Prozeß des Entwerfens selbst kann ebenfalls eine wissenschaftliche Tätigkeit sein, denn wissenschaftliche Methoden und Instrumente sind im Entwurf anwendbar und haben sich aus der Tätigkeit des Entwerfens entwickelt.

In der Informatik gehört zum Beispiel der Begriff 'Computerarchitektur' zum Fachvokabular. Dies kann zu Verwirrung führen, da der Begriff bereits von Seiten der Architektur mit Bedeutung belegt ist. Auch ist bekannt, daß beim Programmieren von Computern Kenntnisse des architektonischen Entwerfens von großem Nutzen sein können. Damit ist der zweite Schwerpunkt dieses Kapitels bestimmt, die Beschreibung von Methoden und Instrumenten in der Architektur, die mit wissenschaftlichem Vorgehen kompatibel sind. Zu Beginn steht eine kurze Beschreibung der Verwendung traditioneller Medien im Entwurfsprozeß.

Traditionelle Entwurfsprozesse - Medium Papier

Das Wort traditionell umschreibt den Entwurfsprozeß, wie er vor der Einführung von Computern in die Architekturschulen und in die Architekturpraxis gelehrt wurde. Dies bedeutet, daß die hier beschriebene Situation in dieser Form kaum noch anzutreffen ist, da zu Beginn der neunziger Jahre Mischformen zwischen traditioneller und computerbasierter Ausbildung und Praxis zur Regel geworden sind. Die Curricula der meisten Architekturfakultäten waren durch die Gliederung in zwei große Gruppen gekennzeichnet: Entwurf und andere Fächer meist technischer Art. Die Einführung in das Entwerfen geschah entweder vor (Beispiel Carnegie Mellon University), während (Beispiel ETH Zürich) oder nach (Beispiel TU München) der Vermittlung mathematischer, technischer, rechtlicher und sozialer Aspekte der Architektur. Während für die technischen Fächer Skripten, Lehrbücher und Prozeduren zur Lösung von Aufgaben zur Verfügung standen, versuchten Architekturlehrende den Studierenden das Entwerfen in Form des 'Studios' zu vermitteln, für das sich verschiedene Formen entwickelt haben, das aber im Grunde überall durch ein intensives Engagement zwischen Meister und Schüler geprägt war. Diese oft sogar persönlichkeitsbildende Beziehung war ein wichtiges Merkmal der Architekturausbildung. Die Architekturlehrenden waren oft zugleich im eigenen Büro tätig, was der Lehre eine praktische Ausrichtung gab; oder sie konzentrierten sich nur auf die Ausbildung unter gewolltem oder erzwungenem Verzicht auf die eigene Praxis, was der Lehre eine eher theoretische Richtung gab. Ersteres herrschte in Mitteleuropa vor. Letzteres fand sich oft in den nordamerikanischen Schulen und ging mit einem sehr günstigen Zahlenverhältnis zwischen Lehrenden und Lernenden einher. Das ungünstigere Zahlenverhältnis zwischen Lehrenden und Lernenden wurde durch die Einschaltung der

Zwischenebene der Assistentinnen und Assistenten gelöst. Viele der bekannten Architekten dieses Jahrhunderts verbrachten die späteren Jahre ihrer Karriere zumindest zeitweise als Architekturlehrer und dienten nachfolgenden Generationen als Vorbild.

Die Rolle der Theorie und der Wissenschaftlichkeit spielte im Entwurfsunterricht mit einigen Ausnahmen in den siebziger Jahren eine oft untergeordnete Rolle. Punktuellen, klar definierten Vorgehensweisen stehen große Gebiete undefinierten Territoriums gegenüber, in denen Studierende ihren eigenen Weg zum Entwerfen finden mußten. Oft gab es zwischen den Mitgliedern einer Entwurfsjury große Meinungsverschiedenheiten, was die Studierenden aus den Beurteilungen anderer Fächer nicht kannten.

Arbeitsinstrumente waren Konstruktions- und Zeichengeräte. Der Zeichentisch war der Arbeitsplatz, an dem der Entwurf entwickelt und für Präsentationen auf Papier übertragen wurde. Die Arbeitsumgebung in Architekturschulen spiegelte diejenige in Architekturbüros wider. Umfangreiche Sammlungen an Materialproben und Baugesetzen wurden von verschiedenen Mitarbeiterinnen und Mitarbeitern im Büro gemeinsam genutzt. In Büros kamen noch Kataloge von Bauteilen hinzu, die für die endgültige Spezifikation von Materialien, Details, Möbeln und anderen Elementen dienten. Seit spätestens Mitte der achtziger Jahre beginnt der Computer die Praxis und die Lehre zu verändern. In der innovativen Verwendung des neuen Mediums übernehmen Praxis und Lehre abwechselnd die Führung, was auch den Entwurfsprozeß der führenden, bisher mit traditionellen Medien arbeitenden Schulen verändert [Burns 1993]. Damit bezeichnet der Anfang der neunziger Jahre die Zeit, in der dem Entwerfen mit Hilfe des Computers zum erstenmal eine ernsthafte Chance gegeben wurde.

Wissenschaftliche Entwurfsprozesse - Design Methods

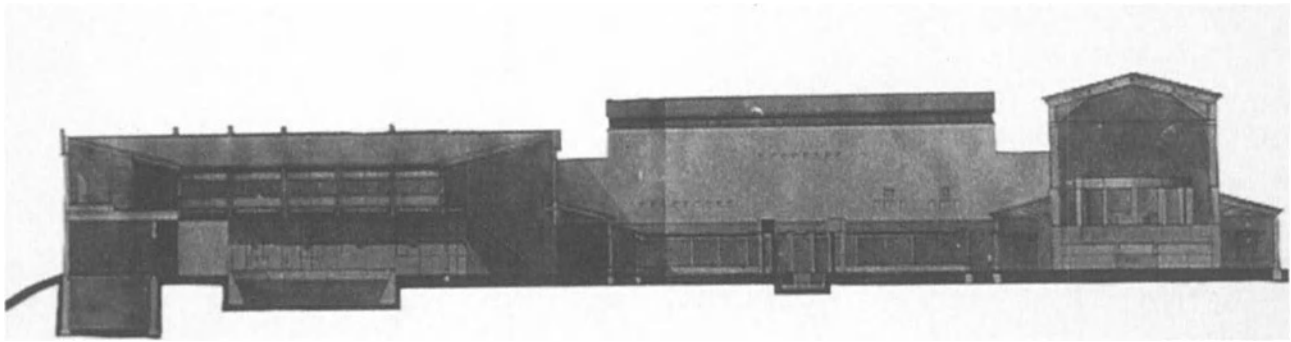
Die wissenschaftliche Betrachtung des Entwurfsprozesses und die anschließenden Versuche der praktischen Umsetzung hatten ihre Blütezeit in den späten sechziger und in den siebziger Jahren. Zunächst bekämpft, danach kaum noch ernst genommen, sind doch einige der damals publizierten Ideen in Praxis und Lehre eingeflossen. Die Hauptkritik an solchen Versuchen war immer der Hinweis, daß sich das im ursprünglichen Sinne des Wortes architektonische Denken und Erfinden der Formalisierung entziehe. Kernbereiche der Architektur, wie Konzeptfindung und -umsetzung, ließen sich, sagte man, nicht durch Programme ersetzen. Der Versuch der Formalisierung des Entwurfsprozesses führte zum Beginn der Entwurfsmethodik- (Design Methods) Bewegung, für die wissenschaftliche Modelle bestimmend waren. Ein wiederkehrendes Thema ist dabei der Wunsch, Design Methods in eine Science of Design weiterzuentwickeln. Die deutsche Übersetzung dafür ist am ehesten Entwurfswissenschaft oder wissenschaftliches Entwerfen. Zwei Theorien erlangten zumindest im englischsprachigen Raum einen gewissen Bekanntheitsgrad in der Architektur: Christopher Alexanders 'Notes on the Synthesis of Form' [Alexander 1964] und Herbert Simons Ansatz in 'The Science of the Artificial' [Simon 1982]. Zu Beginn der neunziger Jahre gingen Roozenburg und Cross rückblickend in einer detaillierten Diskussion auf die methodische Erforschung des Entwurfsprozesses ein [Roozenburg 1991]. Sie beschrieben wesentliche Unterschiede zwischen dem Entwurfsprozeß in den Ingenieurwissenschaften und in der Architektur. Während in den Ingenieurwissenschaften ein allgemein akzeptiertes Konsensmodell des Entwurfsprozesses existierte, gab es dazu im Architekturbereich keine

direkte Entsprechung. Nach Roozenburg und Cross verlief die Entwicklung in folgenden Stufen:

Anfang der sechziger Jahre benutzten Jones und Thornley sehr ähnliche Modelle, um den Prozeß des Ingenieurentwurfs und des architektonischen Entwurfs zu charakterisieren. Sie publizierten ihre Sicht der Dinge auf einer Konferenz über Entwurfsmethoden [Jones 1963]. Im Ingenieurbereich entwickelte sich daraus ein Konsensmodell, das seinen Niederschlag sogar in einer VDI-Richtlinie fand [VDI 1973]. Jones und Thornley sahen den Entwurf als geordneten Prozeß mit den Phasen Spezifikation, Funktionsstruktur, Gesamtlösung, Einzellösung, Vorentwurf, Entwurf und Dokumentation.

Bereits Mitte der sechziger Jahre wehrte sich Christopher Alexander gegen die Sichtweise, daß das Gesamtproblem beim Entwerfen in verschiedene Unterprobleme zerlegbar sei, und daß ein neuer Entwurf zwangsläufig aus der Lösung der Teilprobleme entstehe [Alexander 1964]. Schlagwortartig formulierte er dies in seinem Artikel 'A city is not a tree' (Eine Stadt ist kein Baum) und erschütterte so das wohlgefügte Fundament der vorherrschenden Theorie [Alexander 1966]. Alexander, ursprünglich Mathematiker, schockierte die Architektenwelt mit seinem Vorschlag, intuitiven Individualismus durch rationale, explizite Entwurfsmethoden zu ersetzen. Er schlug die Anwendung von Set- und Graphentheorie, von Klassifikation und strukturellen Beziehungen vor. Er bezeichnete als Beginn der Designanalyse eine Forderung, als Endpunkt das Bauprogramm in Form einer Hierarchie von Forderungssets. Als Anfangspunkt der Designsynthese sah er das Diagramm, als Ergebnis das Erkennen des Problems in Form eines Baumes von Diagrammen. Alexander unterschied zwei Entwurfsprozesse, den

Versuch der Anwendung wissenschaftlicher Prinzipien im Entwurf: Kartause in Spanien. Aquarell, Schmitt, 1979

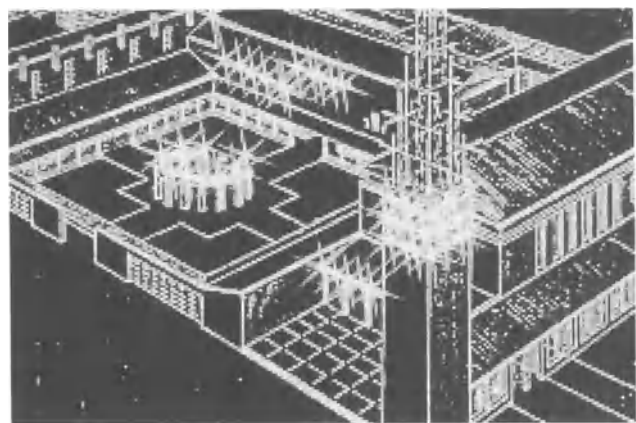
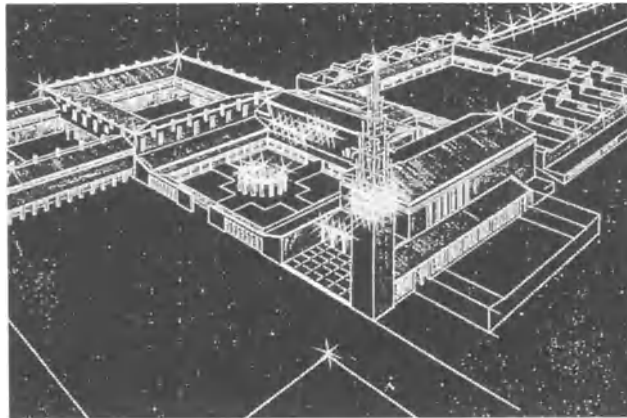


des 'unbewußten' und den des 'bewußten'. Der 'unbewußte' Prozeß dominierte nach seiner Ansicht in primitiven Kulturen und in der 'volkstümlich' (vernacular) genannten Architektur. Den 'bewußten' Prozeß sah er als typisch für professionell ausgebildete Designer und Architekten an. Alexander schien eine Vorliebe für den 'unbewußten' Ansatz zu haben. Dieser Ansatz kann im Gegensatz zur traditionell oft angewandten Top-Down-Methode als Bottom-Up-Methode gesehen werden (siehe Methode: Bottom-Up) [Grabow 1983].

In den frühen siebziger Jahren nahmen die Gemeinsamkeiten in der Betrachtung von architektonischem und ingenieurmäßigem Entwerfen ab. Besonders Hillier kritisierte die Auffassung, daß Architekten sich dagegen wehren sollten, eigenes Wissen und fixe Vorstellungen in den Entwurf mit einzubringen und von Anfang an zu verwenden [Hillier 1984]. Das Hauptargument war, daß Entwerferinnen und Entwerfer die Probleme vorstrukturieren müßten, um zu einer Lösung zu kommen. Hillier zufolge beruhte das im Ingenieurbereich verwendete Analyse-Synthese-Modell, bei welchem jedem Syntheseversuch eine ausführliche Analysephase vorausgehen mußte, auf einer falschen Einschätzung der Rolle induktiver Logik. Demgegenüber

schlug er ein Modell vor, bei dem zu Beginn eine Annahme (Conjecture) stand, die danach in einem Analyse- und Evaluationszyklus überprüft wurde. Ebenfalls in den frühen siebziger Jahren führte Herbert Simon die Unterscheidung zwischen Design und Wissenschaft ein. Die Wissenschaft habe sich damit zu beschäftigen, wie die Dinge sind, Design jedoch damit, wie die Dinge sein sollten. Er ging von einem hierarchisch organisierten Designprozeß aus und von der Annahme, daß der Prozeß fast vollständig in Unterprobleme zerlegbar sei. Ein zunächst unvollständig definiertes Entwurfsproblem ließ sich damit in ein gut strukturiertes Problem überführen, auf das dann ein allgemeines Problemlösungsverfahren (General Problem Solver, GPS) anwendbar war. Der Hauptunterschied zwischen gut und schlecht strukturierten Problemen wurde damit auf die Größe der Wissensbasis reduziert. Clancey schlug eine wichtige Erweiterung des Modells vor und führte zwei Hierarchien ein, die erste für Anwendungsprobleme, die zweite für allgemeines Problemlösen [Clancey 1986]. Er unterschied zwischen Analyseproblemen - Identifikation von Objektmengen nach deren Eigenschaften - und Syntheseproblemen - Zusammenstellung einer Lösung aus Komponenten oder Lösungen aus

Wirtschaftsgebäude, Brüderhaus,
Hauptgebäude, Refektorium,
Bibliothek, Kirche, und Mönchszellen
der Kartause in Nordspanien

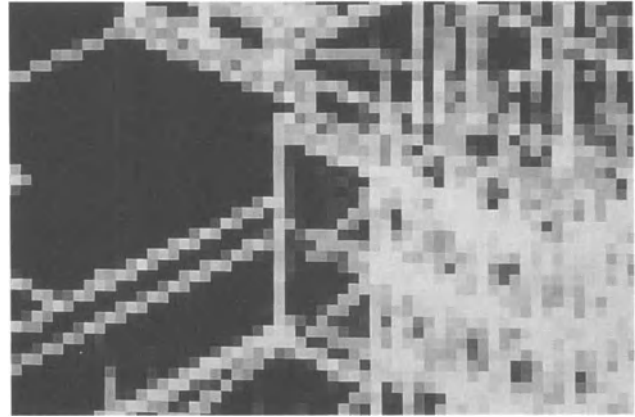
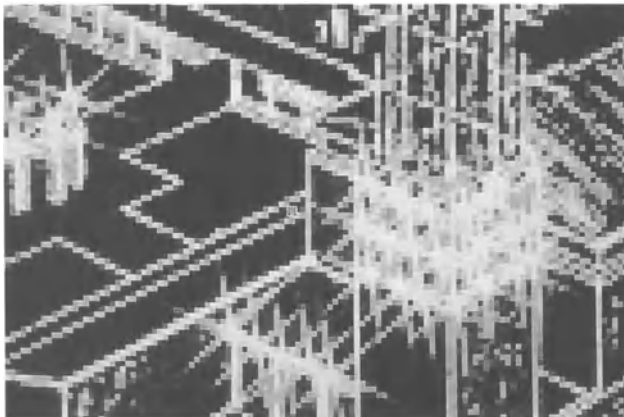


Unterproblemen. Nahe verwandt mit diesem Modell war auch die Information Processing Theory von Newell und Simon [Newell 1972]. Danach ist der Mensch ein Informationsverarbeitungssystem, das aus einem aktiven seriellen Symbolmanipulationsprozessor, sensorischem Aufnahmesystem, motorischem Ausgabesystem, internem Kurz- und Langzeitgedächtnis, sowie externem Gedächtnis besteht. Jeder Problemtyp wurde durch einen symbolischen Suchraum repräsentiert; Problemlösen bestand im systematischen Suchen durch die alternativen Möglichkeiten. Effiziente menschliche Problemlöser reduzierten danach ihren Suchraum auf ein dem Problem angemessenes Maß. Die Struktur des Suchraumes bestimmte die Natur des Computerprogramms, das zur Problemlösung eingesetzt werden konnte. Auch vorher erworbenes Wissen und Erfahrung spielten eine entscheidende Rolle. Der Architekt Ömer Akin erweiterte diese Theorie auf die Architektur [Akin 1986]. Nach seiner Auffassung nutzte ein Entwerfer einen zielorientierten Weg zu einer Lösung, die verschiedene Kriterien erfüllte, welche entweder existierten oder

unterwegs etabliert werden mußten. Dazu wurde eine generative Strategie benutzt, die nach Alternativen suchte und diese dann mit den relevanten Kriterien verglich. Akin entwickelte verschiedene Prototypen, um die Theorie zu testen.

In den frühen achtziger Jahren entwickelte Darke einen weiteren Ansatz. Durch Interviews mit Architekten kam sie zu dem Schluß, daß es ein Generator-Annahme-Analyse-Modell geben müsse. Generator bedeutete dabei die erste, starke Idee, mit der eine sehr beschränkte Zahl von Entwurfszielen einherging [Darke 1984]. Dies stand natürlich im Gegensatz zu der Annahme, daß jedem Entwurfsprozeß eine ausführliche Analysephase vorausgehen müsse. Vielmehr beruhten Lösungskonzepte auf vorstrukturierten Entwurfsmodulen. Lionel March zufolge verlangte die essentielle Logik des Entwurfsprozesses, daß Entwurfskonzepte durch produktives und nicht durch induktives oder deduktives Schließen entstehen [March 1984]. Dementsprechend bestand der rationale Entwurfsprozeß aus (1) der Kreation einer neuen Komposition durch produktives Schließen, aus (2) der Voraussage der Ver-

Weiterbearbeitung einer
 Photographie des Papier-Originals:
 Inversion und stufenweise
 Vergrößerung. Schmitt, 1989



haltensevaluation durch Deduktion und (3) aus dem Zusammentragen von Gewohnheiten und Erfahrungen in eine sich entwickelnde Typologie durch Induktion. Daher der Name des Modells: P-D-I (Produktion, Deduktion, Induktion). Parallelen zum Modell von Hillier waren unübersehbar.

Insgesamt ließen sich einige Gemeinsamkeiten zwischen diesen Theorien feststellen. Alle gingen von einer sorgfältigen Analyse des Problems aus und stellten danach Beziehungen zwischen den einzelnen Faktoren her, so daß alle Unterprobleme identifizierbar wurden. Die Beschreibungen stellten die spiralförmige, zyklische Charakteristik des architektonischen Entwurfsprozesses heraus, während der Ingenieurentwurf anscheinend geradlinig und sequentiell verlaufen sollte. Alle Theorien versuchten, trotz Vorbehalten bei Alexander, die Teilung des Gesamtproblems in Unterprobleme und danach die Erzeugung einer Gesamtlösung durch die Kombination partieller Lösungen. Alexander, Akin, Eastman, Newell und Simon hatten intensiv mit dem Computer gearbeitet. Allerdings wurde keine der Theorien genügend tief implementiert, um eine

praktische Nachprüfbarkeit ihrer Gültigkeit zu ermöglichen.

Auch gegen Ende der Design Methods-Bewegung blieb die Frage offen, warum die meisten der 'großen' Architekten nicht in der Lage waren, ihren Entwurfsvorgang eindeutig darzustellen, nachvollziehbar zu machen und zu externalisieren, während diejenigen, die dies überzeugend tun konnten, keine 'großen' Architekten waren oder wurden. Heute trifft man immer wieder Vertreter der Generation der Design Methods-Bewegung, die die neuen Entwicklungen auf dem Gebiet des CAAD mit Erstaunen betrachten und sich wundern, daß so viele Dinge nach nur 20 Jahren wiederentdeckt werden. In der Tat kehren viele der damals entwickelten Ideen über den Umweg der Computerwissenschaften und der Künstlichen Intelligenz (KI), diesmal mit höherem wissenschaftlichen Anspruch, in die Architektur zurück. Gleichzeitig muß der aus dieser Zeit stammende große Unterschied zwischen intellektuellem Anspruch und gebautem architektonischem Resultat als Warnung für CAAD-Begeisterte dienen.

Neue Entwurfsprozesse - Medium Computer

Der neue, computerbasierte Entwurfsprozeß ist Gegenstand systematischer Forschung in Architektur [Schmitt 1992b, S. 13] und Psychologie [Akin 1986]. Doch selbst bei Beherrschung eines Entwurfsprozesses bedeutet dies noch nicht, daß die so entstehenden Entwürfe auch zur Ausführung kommen. Schon heute ist es sicherlich so, daß in den industrialisierten Ländern weitaus mehr Projekte als wirkliche Architektur erzeugt werden. Der Prozeß der Entstehung von Architektur ist demnach sehr wichtig und oft auch das einzige Produkt neben der Zeichnung. Auch nach der Ära der Design Methods gab und gibt es Versuche der Formalisierung von zumindest Teilen des Entwurfsprozesses, diesmal unterstützt durch den Einbezug des Computers. Nach seinen frühen theoretischen Beiträgen lieferte Christopher Alexander mit seiner 'Pattern Language' ein eher praktisches Beispiel geliefert [Alexander 1977]. Alexanders viel kritisierte Vorschläge sind auch heute noch insofern interessant, als er als Mathematiker und mit dem Computer vertrauter Wissenschaftler früh Zugang zu den neuen Medien hatte. Die von ihm entwickelten Patterns (Muster) weisen bereits Parallelen zur Struktur der Wissensmodule auf, die eines der Grundelemente der Künstlichen Intelligenz und deren Anwendung in der Architektur sind. Doch auch für Alexander dauerte es lange, bis die ersten Gebäude in Übereinstimmung mit dem Prozeß entstanden. Und das Entstehen jedes neuen Gebäudes änderte den Prozeß. Am Beginn der Suche nach alternativen grundlegenden Entwurfsmodellen stand aus praktischen Gründen und in der ursprünglichen Euphorie gegenüber der neuen Technik der Wunsch, rationale Entwurfsmodelle auf dem Computer zu implementieren.

Es ist einerseits notwendig, die Vorgänge, die beim Entwerfen erkannt und erklärbar sind, zu formalisieren und zu versuchen, sie mit Programmen zu unterstüt-

zen. Andererseits ist es notwendig, Möglichkeiten, die nur der Computer bietet, vorzustellen und diese im Hinblick auf ihre Nutzbarkeit für den Entwurf zu untersuchen -notwendigerweise ein experimentelles Unterfangen, da auf diesem Gebiet erst wenige Erfahrungswerte vorliegen. Ideal wäre deshalb die Schaffung einer Entwurfsumgebung, in der verschiedene computerbasierte Methoden und Instrumente jederzeit zur Verfügung stehen, miteinander kombinierbar sind und an realen Entwurfsproblemen erprobt werden. Bearbeitet wird dabei ein großes, intelligentes Modell der entstehenden Architektur, das gewisse Aktionen zuläßt und andere nicht. Dieses Modell sieht das Gebäude als Organismus. Von einem technischen Standpunkt aus gesehen bedeutet dies, daß ein, in eine bestehende Umgebung eingesetztes Gebäude, sein eigenes Ambiente, sein eigenes Mikroklima schafft und sich so eine Balance mit der Umgebung herstellt. Vom architektonischen Standpunkt aus wird die Sicht des Gebäudes als die eines Organismus ebenfalls unterstützt. So bemerkte Rafael Moneo bei seiner Eröffnungsrede in Harvard: "Wenn Architekten erkennen, daß ein Gebäude sein eigenes Leben meistert, ist ihr Herangehen an den Entwurf ein anderes. Es ändert sich radikal. Unsere persönlichen Belange werden sekundär, und die endgültige Realität des Gebäudes wird zum authentischen Ziel unserer Arbeit." [Lacy 1991, S. 158, dt. v. Verf.].

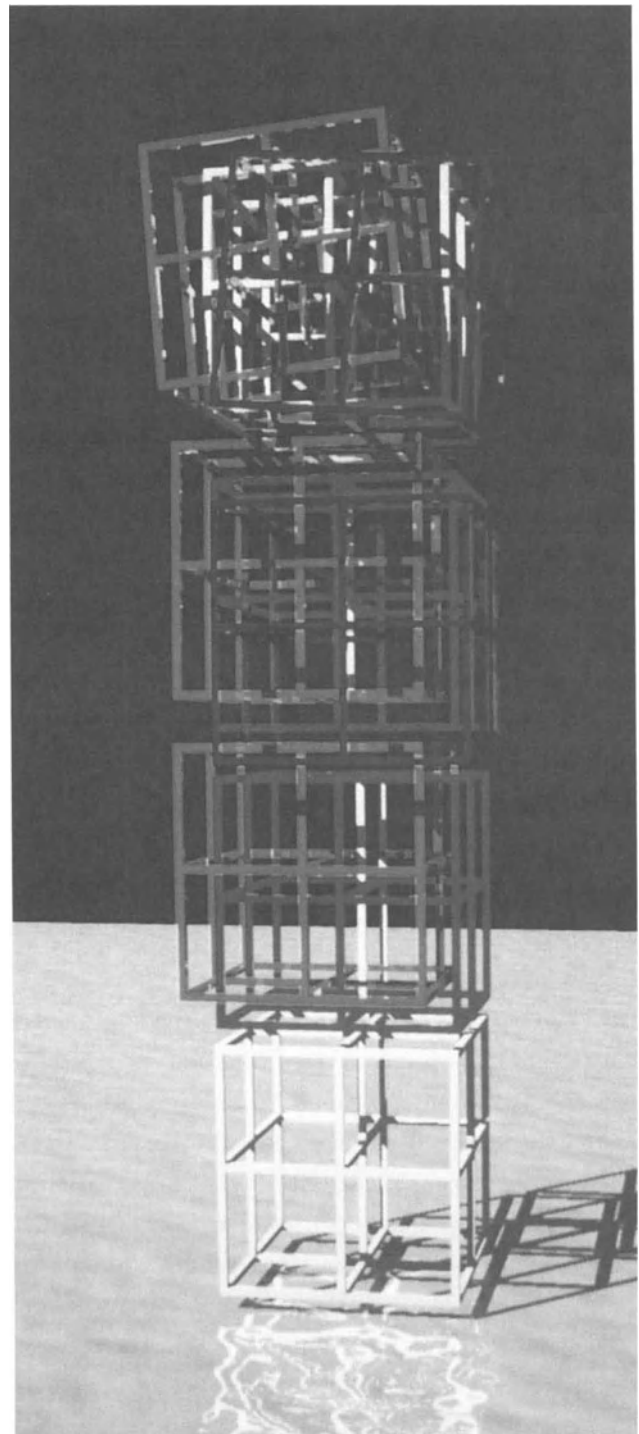
Es ist heute nicht mehr möglich zu definieren, wie ein Entwurfsprozeß abzulaufen hat, denn offensichtlich gibt es viele verschiedene Modelle. Doch ein mögliches Modell der architektonischen Informationsverarbeitung können wir von Herbert Simon übernehmen, das er auch für das kreative Entwerfen als gültig erklärt [Simon 1992]. Vereinfacht ausgedrückt, beschreibt er die verschiedenen im Gedächtnis ablaufenden Klassifizierungs-, Indexierungs-, Such- und Kombinationsme-

chanismen, unterstützt durch nachvollziehbare Vorgänge im Kurzzeit- und Langzeitgedächtnis, als für das Entwerfen ausreichend. Man kann darüber streiten, ob das menschliche Entwerfen wirklich so funktioniert, aber das soll hier weniger interessieren. Wichtig ist die Annahme, daß man sich Teile des Entwerfens als externalisierte Prozedur vorstellen kann. Damit ist sie durch ein Computerprogramm nachvollziehbar. Es ist nicht notwendig, daß die menschliche und die Computerrepräsentation übereinstimmen, aber beide müssen vollständig sein und zum Ziel der Architektur führen.

Am Ende der theoretischen Diskussion bleibt die Frage, wie der Entwurf wirklich funktioniert. Es ist gewiß möglich, durch die sorgfältige Wahl und Kombination der im folgenden vorgestellten Methoden und Instrumente einen Entwurf von der ersten Idee bis zur Ausformulierung zu erarbeiten. Wir haben entsprechende Erfahrungen in der Lehre, Forschung und Praxis gemacht und werden sie mit vielen Beispielen belegen. Doch entwerfen Architekten wirklich so? Zumindest jetzt noch gibt es gewaltige Differenzen in ihrer Auffassung und Beschreibung eben jenes Vorganges, der doch eigentlich am besten bekannt sein müßte. Das Buch von Bill Lacy, das Skizzen, Beschreibungen und Projekte von 100 bekannten Architektinnen und Architekten vorstellt, ist in dieser Beziehung sehr aufschlußreich [Lacy 1991], ebenso die in den verschiedenen Architekturjahrbüchern vorgestellten Projekte [Klotz 1989, Neitzke 1992].

Der Entwurf ist ein Prozeß, den wir weitgehend selbst bestimmen können. Mit neuen Methoden und Instrumenten als gedanklichen und computerimplementierten Werkzeugen sollten auch bessere Resultate möglich werden. Die Koexistenz zwischen computerbasierten und intuitiven Entwurfsmethoden hat damit begonnen.

Objekt entstanden aus der Kombination von Regeln und Intuition: Tower of Thought. Grohe, 1991, Visualisierung Refvem, 1991



Repräsentation von Architektur

In den vorangegangenen Abschnitten wurden verschiedene Ansätze für die Formalisierung und Vermittlung von Entwurfsprozessen behandelt. Doch der Begriff Architektur umfaßt neben den Prozessen auch physische Objekte mit unterschiedlichsten Eigenschaften, Funktionen, Beschränkungen und Beziehungen. Um sie angemessen beschreiben und verarbeiten zu können, sind Darstellungsformen oder Repräsentationen notwendig, durch die wir unsere Vorstellungen über Architektur vermitteln können. Mit der Einführung des CAAD in Lehre und Praxis wird die Bedeutung der Repräsentation von Architektur in ihrer Gesamtheit als fundamentale Größe noch offensichtlicher. Repräsentation ist etwas so Grundlegendes, daß sie ebensowenig in Kategorien wie in Methoden oder Instrumente einzuordnen ist.

Repräsentation ist die vereinfachte, doch wesentliche Eigenschaften umfassende Darstellung eines Objektes oder Zustandes zum Zweck der Übermittlung und Weiterverarbeitung. Datenrepräsentation ermöglicht die Beschreibung von Problemen und Teilproblemen als Objekte sowie deren Beziehungen untereinander im Computer. Wenn sich zwei Menschen über einen Gegenstand unterhalten, so müssen beide über eine interne Repräsentation dieses Gegenstandes verfügen. Der Grad der Exaktheit der Repräsentation entscheidet über die Qualität der Aussagen, die über den Gegenstand gemacht werden können. Wenn Architekten über Architektur sprechen, so gehen sie dabei von einer anderen Repräsentation aus als zum Beispiel ein Haustechniker oder eine Politikerin, obwohl sie von demselben Objekt reden. Noch größer werden die Unterschiede beim Versuch der Integration von Computern in den Entwurfsprozeß. Nur über gemeinsame oder zumindest kompatible Repräsentationen ist es möglich, zwischen Entwerfenden und der Maschine zu kommunizieren.

Eine zusätzliche Komplikation entsteht dadurch, daß Menschen Computer als Projektionsmaschine benutzen. Sie projizieren Wünsche und Verhaltensweisen auf die Maschine, die auf gewisse Anregungen reagiert, selbst aber nicht über die gleichen Repräsentationen verfügt, von denen die Menschen ausgehen. Wenn Menschen über komplexe Themen wie Architektur reden, kann man zusätzlich davon ausgehen, daß sie eine gemeinsame Sprache sprechen und daß sie ein ähnliches Maß an Allgemeinwissen besitzen. Sobald der Computer in diesen Prozeß miteinbezogen wird, ist weder die Voraussetzung gemeinsamer Repräsentation noch gemeinsamer Sprache gegeben. Bestenfalls können syntaktische Vereinbarungen definiert und eingehalten werden. Das drückt sich dann darin aus, daß beim Befolgen der Syntaxregeln eine fruchtbare Zusammenarbeit möglich ist. Bei Nichtbefolgung dieser Vereinbarung ist eine Weiterarbeit nicht mehr möglich: Der Mensch muß sich dem Syntax der Maschine fügen, oder umgekehrt. Ein Grund dafür ist, daß Programme erst ein geringes Maß an Allgemeinwissen besitzen und noch nicht in der Lage sind, implizite Anforderungen zu verstehen, also semantisch auf menschliche Wünsche zu reagieren. Ausnahmen sind scharf definierte Teilgebiete, in denen entweder der Mensch mit der Repräsentation der Maschine arbeiten kann oder umgekehrt. Weitere Ausnahmen sind Computerprogramme und -komponenten, von deren Funktionen das Leben vieler Menschen abhängt. Diese Programme und Komponenten müssen fehlertolerant sein und gewisse semantische Interpretationen im Not-situationen vornehmen. Doch auch dies funktioniert nur, wenn der Maschine eine möglichst exakte Repräsentation der möglichen Fälle zur Verfügung steht. Programmieren ist das Instrument, mit dem der Mensch der Maschine exakte Anweisungen gibt, die vom Computer verstanden und umgesetzt werden. Dabei ist zu

beobachten, daß sich mit jedem Schritt in der Software-Entwicklung zusätzliche Schichten zwischen Anwender und Maschine schieben, die wiederum ihre eigenen Repräsentationen beinhalten.

Für die weitere Entwicklung des CAAD in Richtung eines benutzerfreundlichen Instruments ist es unabdingbar, daß die menschliche Repräsentation von Objekten und Funktionen sowie die Computerrepräsentation derselben Objekte und Funktionen einander ver-

stehen oder zumindest miteinander kompatibel sind. Zu erreichen ist dies mit den im nächsten Abschnitt beschriebenen Methoden der Abstraktion und der Modellbildung. In der Literatur findet sich eine gute und immer noch gültige Zusammenfassung der verschiedenen Repräsentationen von Geometrie und Funktion von Gebäuden in dem Buch 'Computer-Aided Architectural Design' [Mitchell 1977]. Eine andere, weitergehende, aber noch immer nicht die Essenz der

Vergleichende
Repräsentationen von
Mensch und Maschine.
Dave, 1993

Mensch

Konzept. Beispiele: Berg, Intelligenz;
alle Konzepte sind miteinander verknüpft

Relation. Beispiele: Beziehungen zwischen Konzepten, die einen hohen Komplexitätsgrad erreichen können

Verhaltensmuster. Beispiele: Eingespielte Vorgehensweise

Vernetztes Denken.

Beispiele: Inbeziehungsetzen von Konzepten

Rahmen. Beispiele: Konzeptstrukturen, in denen zusammengehörende Informationen gespeichert sind

Wissen und Erfahrung. Beispiele:
Wissen, das reflexhaft zu richtigem Verhalten führt

Konventionen, Regeln.

Beispiele: Allgemein akzeptiertes Verhalten

Problemlösung.

Beispiele: Zerlegen in Teilprobleme, zielgerichtetes Vorgehen mit Zwischenschritten

Entwurfsdenken.

Beispiele: Architektur in verschiedenen Kulturen

Qualitative Geometrie.

Beispiele: Ungefähre, aber extrem effiziente Vorstellung geometrischer Objekte für Manipulation und Entwurf

Computer

Objekte. Beispiele: Tisch, Verkabelung; Objekte sind nur punktuell verknüpft, kein Verständnis der Zusammenhänge

Mathematische Operatoren.

Beispiele: and, or, not, >, <, =, <=, >=

Skript. Beispiele: Gespeicherte Abläufe von Instruktionen

Semantische Netze.

Beispiele: Erzeugung von Beziehungen zwischen Objekten

Rahmen. Beispiele: Datenstrukturen in C, Records in Pascal, Association Lists in Lisp

Algorithmen. Beispiele:

Externalisierung und Formalisierung von Abläufen

Regeln.

Beispiele: Wenn-Dann-Regeln, Formen-Regeln

Graph. Beispiele: Zielgerichtetes Vorgehen, das nur aus sequentiellen, konditionellen, sowie Schleifen-Strukturen besteht

Effizienzsteigerung. Beispiele:

Suchraum-Erweiterung, Erzeugung von Alternativen

Digitale Geometrie.

Beispiele: Genaue, weniger effiziente Repräsentation geometrischer Objekte für die Dokumentation

Architektur miteinschließende Art der Repräsentation beschreiben Akin und Weinel [Akin 1982]. Es ist interessant, verschiedene Arten der Repräsentation auf menschlicher Seite mit bereits bestehenden Entsprechungen auf Computerseite zu vergleichen. Diese bilden keinen Gegensatz; vielmehr waren - und sind - die Computerrepräsentationen ein spezialisierter Teil der menschlichen Repräsentationen.

Zusammenfassend und vereinfachend ist die Aussage der Tabelle auf der vorhergehenden Seite so zu interpretieren, daß die menschlichen Repräsentationen qualitativ, kontinuierlich und sensorisch sind, die Computerrepräsentationen dagegen quantitativ, diskret und digital. Doch ist es so, daß jedesmal, wenn wir in der Lage sind, eine Eigenschaft der menschlichen Repräsentationen genau zu definieren, auch ähnliche Computerrepräsentationen gefunden oder entwickelt werden können. Das bedeutet, daß sich die beiden Repräsentationen im Laufe der Zeit annähern und weiterentwickeln werden, was zu einer besseren Zusammenarbeit zwischen den ungleichen Partnern Mensch und Maschine führen wird. Die Maschine kann bei der Definition und Findung von Qualität helfen, indem sie quantitative Evaluationen beisteuert und Architektur in verschiedenen Repräsentationen zeigt, die jeweils einen Aspekt des Gebäudes darstellen. Denn man erkennt Dinge oft erst dann, wenn man die Repräsentation ändert.

Wesentlich schwieriger ist es, in der Maschine eine Repräsentation und damit ein 'Verständnis' von Architektur und ihrer Qualität zu erzeugen, da hier selbst die Menschen sich nicht einig sind. Der Grund ist, daß Programme noch zu wenig von einem Entwurfsproblem in seiner Gesamtheit verstehen. Im Gegensatz dazu sind bei den Menschen verschiedene Repräsentationen intelligent miteinander vernetzt und kompatibel.

Nehmen wir das naive Bild von der Arbeit eines Steinmetzen: Er verwendet eine Reihe verschiedener Methoden und Instrumente, um, von rohem Stein ausgehend, die endgültige Skulptur zu erzeugen. Dabei hat er eine Vorstellung des Endergebnisses ständig vor seinem geistigen Auge, er simuliert also die Skulptur in seinem Gedächtnis und bearbeitet den Stein zielgerichtet. Wichtig ist, daß er, jeden Zwischenschritt mit der Simulation des Endzustandes ständig vergleichen kann. Wie in der Architektur wird er dazu ein Modell und Zeichnungen zu Hilfe nehmen, oder die Arbeit an der Skulptur mag Rückkopplungen ergeben und das Endergebnis mit beeinflussen; aber stets setzt er die Methoden und Instrumente basierend auf seiner eigenen Vorstellung, seiner eigenen Repräsentation des Problems ein.

Oder nehmen wir das Beispiel eines traditionellen Bauarbeiters, der Stein für Stein nach Plan legt. Der Plan hat keinerlei Verständnis für das Gebäude, aber doch repräsentiert er dessen geometrische Ausformung so, daß die Beteiligten meistens eindeutige Interpretationen und Handlungen daraus ableiten können. Wiederum besteht eine Gesamtrepräsentation des fertigen Gebäudes wahrscheinlich nur im Gedächtnis der Architektin oder des Architekten, doch hindert das meist nicht an einer tadellosen Ausführung des Auftrags durch die einzelnen Beteiligten. Möglich wird dies durch die Kompatibilität der Repräsentationen, obwohl das gemeinsame Medium Papier keinerlei Wissen, keine Modellvorstellung hat, nicht mit anderen Dokumenten kommunizieren kann und sich deshalb lediglich eine externe Repräsentation eines Teils des Gedankengebäudes im Kopf der Entwerfenden findet. Hier liegt der große Unterschied zwischen der herkömmlichen Repräsentation und der Repräsentation im Computer. Mit dem Computer ist es plötzlich möglich, wenigstens rudimentäre Beziehungen zwischen den Elementen und Darstellungen herzustellen,

also einen Teil der Gesamtrepräsentation des Gebäudes in die Maschine zu verlegen. Ist dies erst einmal geschehen, kann man der Maschine auch gewisse Entscheidungen zubilligen. Voraussetzung ist, daß das Programm alle wichtigen Regeln und Zusammenhänge kennt und somit keinen Unsinn produziert. Wie die verschiedenen Schach-, Mühle- und anderen Spielprogramme zeigen, ist es nicht unbedingt notwendig, daß die Maschine über dieselbe Repräsentation verfügt wie der Mensch, um hervorragende Leistungen zu erbringen. Die Repräsentation des Schachbretts und der möglichen Züge sieht für das Computerprogramm radikal anders aus als für den Menschen. Der sinnvolle Einsatz des Computers im Entwurf verlangt also nicht dieselbe Repräsentation, sondern eine, die zum selben Ziel führt. Darauf sind die in der Folge beschriebenen Instrumente und Methoden ausgerichtet.

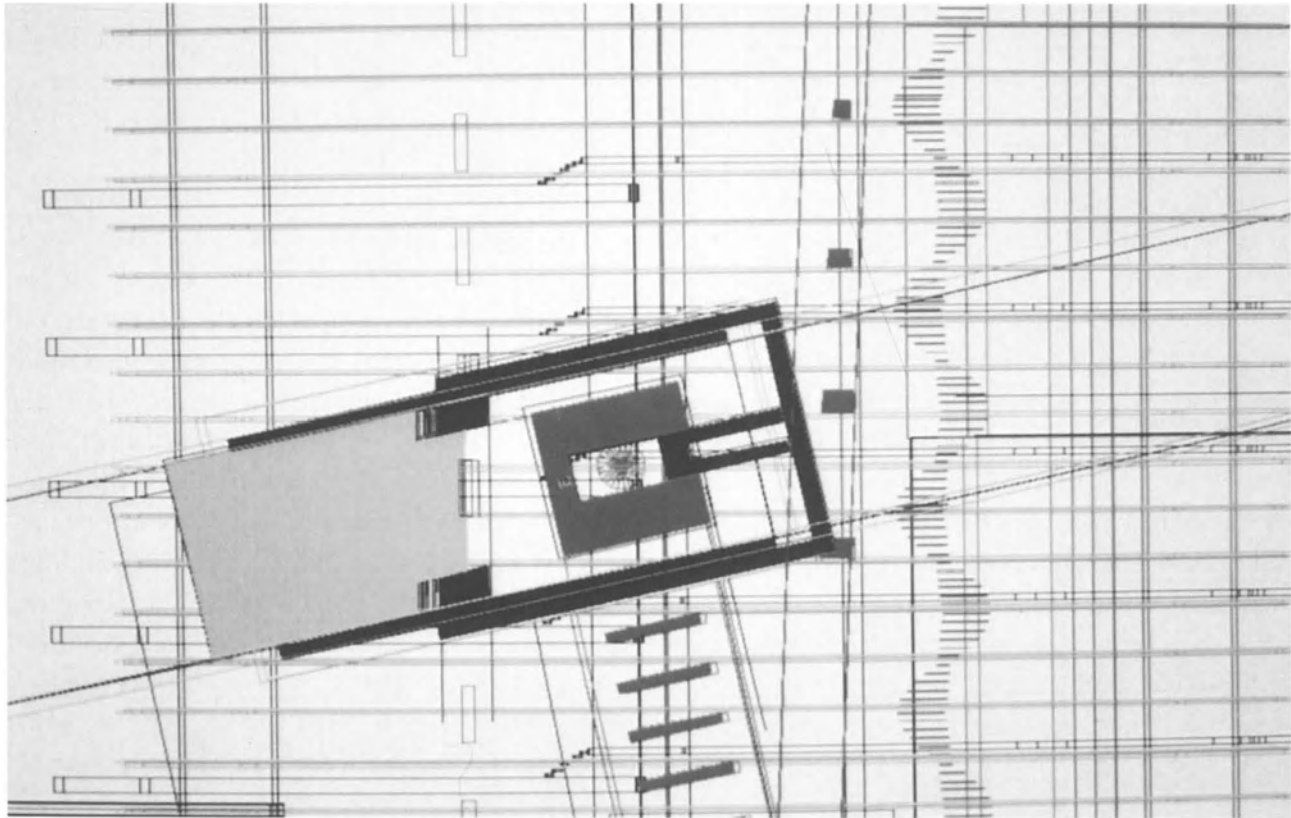
Es besteht das Dilemma, daß die meisten heute bekannten Repräsentationen für menschliche Problemlösungen optimiert sind. Für den Einsatz in der Maschine sind sie also wegen der verschiedenen Strukturen der Wissensverarbeitung denkbar ungeeignet. Bei der folgenden Beschreibung der Methoden und Instrumente soll deshalb im Gedächtnis bleiben, daß der Sinn des Computers nicht die Imitation des Menschen ist, da die Maschine dazu wegen der Andersartigkeit ihres Aufbaus und ihrer Fähigkeiten unbrauchbar ist. Folglich sollte man auch nicht versuchen, nur menschliche Fähigkeiten zu kopieren und auf der Maschine zu simulieren. Vielmehr sind die Besonderheiten der im Computer möglichen Repräsentationen auszunutzen, um Lösungsvorschläge wesentlich schneller und akkurater als der Mensch zu generieren und zu evaluieren. Daneben erlauben es die abstrakten Repräsentationen natürlicher Phänomene, bisher

unsichtbare Dinge sichtbar zu machen. Computer gewinnen dadurch die Fähigkeit, dem Menschen neue sinnliche Erlebnisse zu erschließen.

Die Bandbreite der Computerrepräsentationen reicht von der anwendungsspezifischen, genauen und kleinteiligen, das menschliche Vorstellungsvermögen unterstützenden Darstellung bis zur - noch nicht verwirklichten, aber vorstellbaren - umfassenden Repräsentation eines Gebäudes mit all seinen Funktionen und internen Verknüpfungen. Die Teile eines Gebäudes sind darin genau beschrieben, ebenso ihre Interaktionsmöglichkeiten. Damit sind wir bei den 'intelligenten Objekten', die selbständig miteinander interagieren und Entscheidungen treffen können (siehe Intelligente Objekte). Sie sind die Konsequenz der Ausstattung von Objekten mit künstlichem Wissen. Spätestens an diesem Punkt werden sich die Geister scheiden. Die entsprechende Auseinandersetzung hat auf dem Gebiet der Künstlichen Intelligenz, das sich seit Mitte der fünfziger Jahre mit der Möglichkeit denkender Maschinen beschäftigt, bereits Geschichte. Die Frage danach, ob Computer intelligent sind, interessiert hier wiederum weniger als die Resultate. Das Problem ist nach wie vor kontrovers und scheint Architekten in besonderem Masse zu beunruhigen. Im Gegensatz zu Kritikern der KI wie Dreyfus [Dreyfus 1987] und Searle [Searle 1990] halten andere Wissenschaftler die fortschreitende Erforschung und Nachahmung der Funktionen des menschlichen Gehirns und damit denkende Maschinen für möglich [Churchland 1990]. Die begründbare Annahme, daß alles, was wir heute als Semantik ansehen, in Zukunft als Syntax definierbar sei, gewinnt mit jeder erfolgreichen Simulation von Denkvorgängen an Gewicht - sie beweist aber nicht die Denkfähigkeit der Maschine im menschlichen Sinn.

Methoden für den computerunterstützten Entwurf

Verwendung verschiedener Computermethoden
im experimentellen Entwurf. Grohe, 1991



Methoden entwickeln sich aus generalisierten Erkenntnissen. Einzelne Methoden eignen sich gut für besondere Problemstellungen, wie zum Beispiel die Anpassung von Prototypen für Routine-Entwurfsprobleme oder fallbasiertes Schließen für den Rückgriff auf bereits bestehende, bewährte oder zu vermeidende Lösungen. Kombinationen von Methoden innerhalb eines Systems sind ebenfalls möglich und üblich. Die hier vorgestellten Methoden sind gedankliche Werkzeuge, die den Vorteil haben, daß von ihnen bereits vollständige oder partielle Computerimplementationen bestehen. Methoden, wie sie in der CAAD-Literatur definiert sind, helfen bei der Lösung von Entwurfsproblemen durch die Verwendung von Suchmechanismen. Diese Methoden ermöglichen die Umwandlung komplexer Probleme in einfachere Teilproblembeschreibungen.

Methoden sind problemunabhängig. Die folgenden Methoden sind bereits definiert und haben nach unseren Erfahrungen ihre Nützlichkeit für bestimmte Entwurfsaufgaben bewiesen:

Abstraktion und Modellbildung. Zwei grundlegende Bedingungen für die Kommunikation Mensch-Maschine.

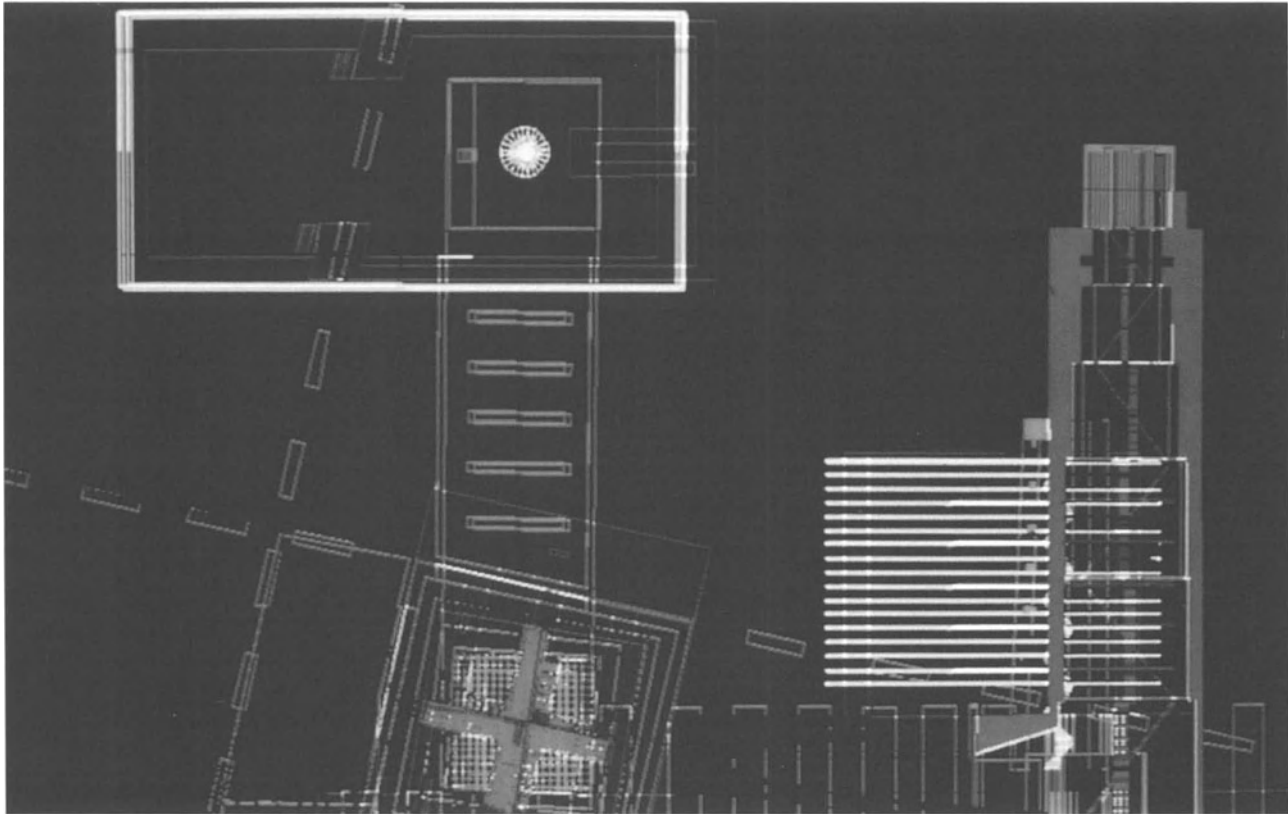
Simulation. Nutzung von Modellen zur quantitativen und qualitativen Darstellung entscheidender Faktoren.

Top-Down. Ableitung einer Gesamtlösung von einer festen Zielvorstellung durch Zerlegung in Teilprobleme.

Bottom-Up. Zusammensetzung einer Gesamtlösung durch schrittweise, iterative oder rekursive Kombination von Einzelelementen.

Prototypen. Anpassung einer parametrischen Objektbeschreibung an die Zielvorstellungen unter der Voraus-

Der Computer als Test- und Präsentationswerkzeug
im experimentellen Entwurf. Grohe, 1991



setzung, daß für das Entwurfsproblem eine prototypische Lösung existiert.

Fallbasiertes Schließen. Nutzung und Aufbau indexierter Fall-Datenbanken, um Fehler der Vergangenheit zu vermeiden und Erfolge zu wiederholen.

Maschinenlernen. Ausnutzung der Fähigkeit von Computern, gewisse menschliche Lernvorgänge und im nächsten Schritt computerbasierte Lernvorgänge zu imitieren.

Die im folgenden beschriebenen Methoden und Instrumente sind Module, die in eine allgemeine Modellierungsumgebung eingesetzt werden können. Wir haben dies innerhalb des Programms AutoCAD exemplarisch realisiert, doch sind auch andere Implementationen vorstellbar. Sie unterstützen verschiedene Aspekte des Planens, Entwerfens und Bauens. Die Liste ist

nicht komplett und wird ständig erweitert. Aber sie repräsentiert bereits jetzt nicht nur eine, sondern verschiedene Vorgehensweisen beim Entwerfen.

Das Format der folgenden Ausführungen gibt den Methoden und Instrumenten jeweils zwei Seiten zur Beschreibung, obwohl dies für den Rang einiger Themen sicherlich nicht ausreicht. Das einheitliche Format jedoch verdeutlicht den instrumentalen Charakter und unterstreicht die Absicht, die Entscheidungsmöglichkeiten explizit zu machen. Der Wunsch nach Explizität verursacht natürlich Vereinfachungen und wahrscheinlich auch einige Kurzschlüsse, kann aber trotzdem Grundlage einer konstruktiven Diskussion sein. Die im Mittelteil des Buches im Kapitel «et» vorgestellten Beispiele zeigen die Anwendung dieser Methoden und Instrumente in Lehre, Praxis und Forschung.

Methode: Abstraktion und Modellbildung

Anwendung: Manipulation und schnelles Entwickeln neuer Anwendungen, Simulation

Verwandte Gebiete: Kunst, Wissenschaft

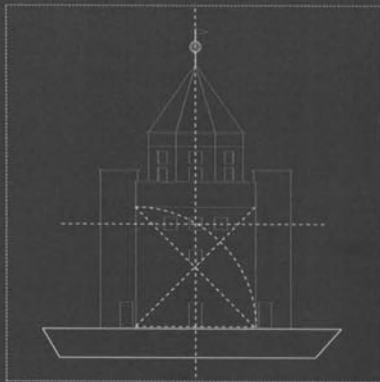
Die Verwendung von Abstraktion als Methode befreit uns von dem Zwang, bei der Darstellung der Wirklichkeit auch Dinge zu repräsentieren, die uns nicht interessieren. Abstraktion verwenden wir immer auch dann, wenn die Realität zu komplex oder zu umfangreich ist, um sie zu erfassen oder zu manipulieren. Realität bedeutet in diesem Fall gebaute oder geplante Architektur. Jeder Versuch, die Wirklichkeit zu repräsentieren, ist bereits eine Abstraktion. Die einzige vollständige Repräsentation der Wirklichkeit ist das Objekt selbst. Abstraktion wird damit zu einer wichtigen Eigenschaft der Repräsentation [Akin 1986, S. 188]. Abstraktion befähigt zur Manipulation von Objekten, zum Entwerfen oder Analysieren neuer Sachverhalte. Die verwendeten Abstraktionen können sich dabei von Phase zu Phase verändern. Es hat sich erwiesen, daß die Benutzung nur einer Abstraktion zum Entwerfen nicht genügt und zu unbefriedigenden Lösungen führt. Ergebnisse sind Gebäude, die zum Beispiel allein auf Energieverbrauch oder Kosten optimiert wurden.

Modelle sind Strukturen, die Abstraktionen und Operanden zu einer für Analyse und Manipulation einsetzbaren Einheit zusammenfassen. Durch Modelle können wir Verhalten, Aussehen und verschiedenste Funktionen eines neuen Gebäudes vereinfacht darstellen und manipulieren und sie Betrachterinnen und Betrachtern vermitteln. Voraussetzung für das Entstehen eines Modells ist das Vorhandensein einer entsprechenden Abstraktion. In Wissenschaft und Architektur ist Modellbildung notwendig, um mit Abstraktionen umgehen zu können. Jedes Modell erfordert eine Reihe von Übereinkünften oder Konventionen, um effektiv damit arbeiten zu können. Je höher der Abstraktionsgrad, desto mehr Konventionen sind einzuhalten.

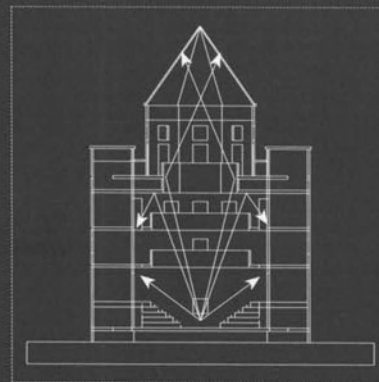
Um Sachverhalte und Objekte aus der realen Welt im Computer verarbeiten zu können, sind Datenmodelle notwendig. Die Computerwissenschaft hat daher eine Reihe abstrakter Datenmodelle entwickelt, mit denen sich Sachverhalte simulieren lassen. CAAD-Modelle sind ein gutes Beispiel, denn in idealer Form verkörpern sie ein Gebäude mit Form, Verhalten und Funktion als logischen und manipulierbaren Organismus. Sie dienen zur integrierten Darstellung verschiedener Aspekte des Planens und Bauens. Konsequenterweise ausgeführt, sind sie vom frühesten Entwurfsstadium bis zum Stadium des Gebäudemanagements einsetzbar.

In der Architektur markierte der Übergang von der direkten Manipulation der Realität - dem Entwerfen und Bauen vor Ort - zur Modellbildung - also der Einführung eines Abstraktionsgrades - einen Durchbruch. Das Modell eines Gebäudes ist heute das eines Organismus. Bekannte Modelle in der Architektur sind das physische Modell und das geometrische Modell. Das physische Modell dient der Vermittlung einer Illusion, die im Betrachter das endgültige Bauwerk als mentales Bild erzeugen soll. Das geometrische Modell benutzt zweidimensionale Abstraktionen, die im geschulten Betrachter dreidimensionale Vorstellungen hervorrufen. Damit ist der Bedarf nach Modellen in der Architektur aber keineswegs befriedigt. Praktisch jede Disziplin hat ihre eigenen Modelle entwickelt, um Teile der Realität zu simulieren. In der Statik sind es Stabmodelle und Finite Elemente, in der Computerwissenschaft sind es dagegen teilweise aus der 'wirklichen' Architektur übernommene Modelle. Im Entwurf von Benutzeroberflächen ist das Modell der Schreibtisch, in neueren Entwicklungen ein Raum oder ein ganzes Gebäude.

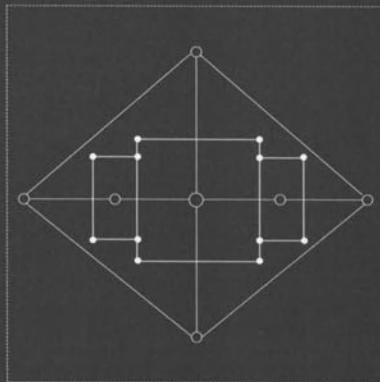
Abstraktion:
Geometrie



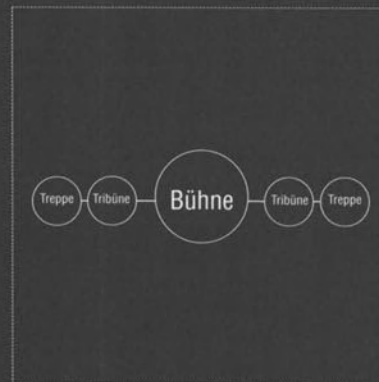
Abstraktion:
Akustik



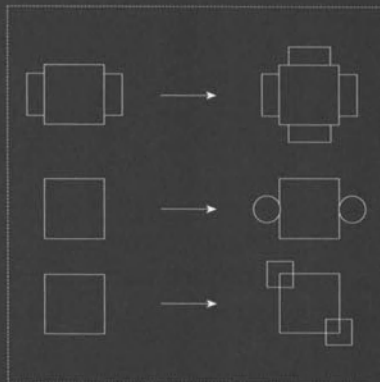
Abstraktion:
Graph



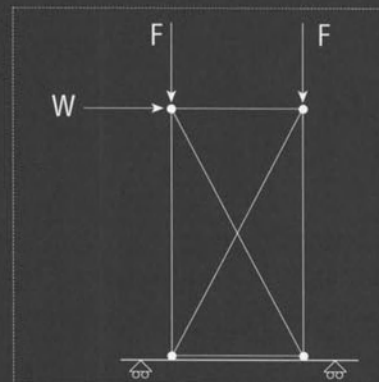
Abstraktion:
Funktion



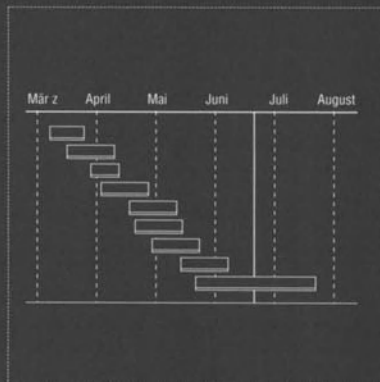
Abstraktion:
Grammatik



Abstraktion:
Statik



Abstraktion:
Bauprozess



Abstraktion:
Beziehungen

	Steg	Schiff	Treppe	Bühne	Tribüne	Terrasse	Balkon
Steg							
Schiff							
Treppe							
Bühne							
Tribüne							
Terrasse							
Balkon							

Van der Mark, 1993

Methode: Simulation

Anwendung: Architektur und Wissenschaft

Verwandte Gebiete: Vergleich, Analogie

Sobald die geeignete Abstraktion gefunden und ein Modell definiert ist, wird eine Simulation möglich. Die Simulation ist zum einen ein Mittel, ein real bestehendes Objekt oder einen real bestehenden Zustand in seinen interessanten Aspekten möglichst wirklichkeitsnah darzustellen. Sie ist zum anderen eine Methode, ein angestrebtes Objekt oder einen angestrebten Zustand in seinen interessanten Aspekten möglichst so darzustellen, daß er mit dem auszuführenden Projekt übereinstimmt. Deshalb muß das Modell möglichst vollständig sein.

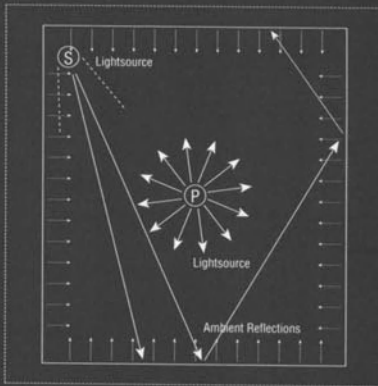
Mit Simulationen lassen sich unliebsame Überraschungen vermeiden. Es ist bekannt, daß Fehler in der Planungs-, Konstruktions- und Nutzungsphase mit drastisch unterschiedlichen Korrekturkosten verbunden sind. Die Angaben reichen von Verhältnissen von 1:10:100 bis zu 1:100:1000 (Kosten der Fehlerbehebung in der Planungsphase im Verhältnis zu den Kosten der Fehlerbehebung in der Konstruktionsphase im Verhältnis zu den Kosten der Fehlerbehebung in der Nutzungsphase). Somit liegt der beste Zeitpunkt für die Simulation möglicher Fehler in der Planungsphase. Um Fehler schon in der 'billigen' Planungsphase vermeiden zu können, muß daher bereits viel Detail- und Allgemeinwissen in die Simulation einfließen. Simulationen dienen auch zur interaktiven Annäherung an einen erwünschten Zustand, so wie es die verschiedenen Suchmechanismen der Künstlichen Intelligenz vorschlagen. Ebenso wie der Planungsprozeß der günstigste Zeitraum zur Simulation mit dem Ziel der Fehlervermeidung ist, so ist dies auch der beste Zeitpunkt zur Simulation der zu erreichenden architektonischen Ziele.

Simulationen sind in allen Bereichen der Architektur und des Entwurfs notwendig. Sie beantworten die Fragen nach den Konsequenzen von Entwurfsentscheidungen. Simulationen haben ein Modell oder mehrere

Modelle als Grundlage und vermitteln das Ergebnis in einer vorher festgelegten Abstraktion. Pläne, Perspektiven oder physische Modelle, die in jedem Gespräch mit der Bauherrschaft gezeigt werden, sind das Ergebnis einer Simulation. Diese graphische Simulation wird ergänzt durch eine Reihe alphanumerischer Simulationen, zum Beispiel auf dem Gebiet der Kosten und des Energieverbrauchs. Die bekannteste Simulation ist die der Gebäudedarstellung. In wechselnden Abstraktionen zeigt sich damit der Entwurf in seinem jeweiligen Stand. Eine weitere wichtige Simulation ist die der Kosten. Interessanterweise sind das Kostenmodell und das CAD-Modell meist noch voneinander getrennt, was zu großem Daten- und Zeitverlust führen kann. Der Energieverbrauch eines projektierten oder bestehenden Gebäudes läßt sich ebenso simulieren [Schmitt 1988a, S. 153], wie das statische System und alle anderen quantifizierbaren Größen. Es lassen sich dabei beliebig viele 'What-If- oder 'Was wäre, wenn'-Szenarien durchspielen, die mit steigender Rechnerleistung je nach Komplexität fast in Echtzeit möglich werden.

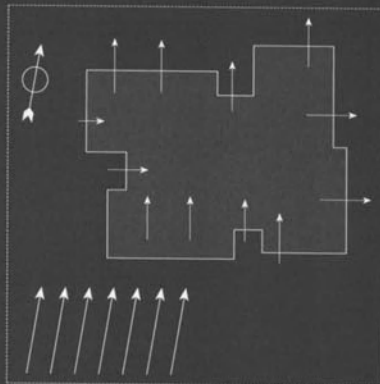
Eine Simulation muß in der Lage sein, der Realität oder Teilen davon sehr nahe zu kommen. Es sollte immer möglich sein, den Abstraktionsgrad einer Simulation zu erhöhen und das Ergebnis in einer anderen oder weniger weit entwickelten Form darzustellen. Die photorealistische Simulation dreidimensionaler Szenen ist ein Beispiel dafür. So lassen sich die abstrakteren Darstellungen wie Flächenschattierungen schnell zur Überprüfung der Massen heranziehen, die photorealistische Darstellung für das Endergebnis. Dieselbe Entwicklung wird sich in der 'Virtual Reality' (VR) abspielen. Ist man jetzt noch mit sehr einfachen Szenen zufrieden, solange man die typischen VR-Aktionen simulieren kann, so wird auch hier sehr bald ein großer Bedarf nach realitätsgetreuer Repräsentation entstehen.

Licht: Abstraktion



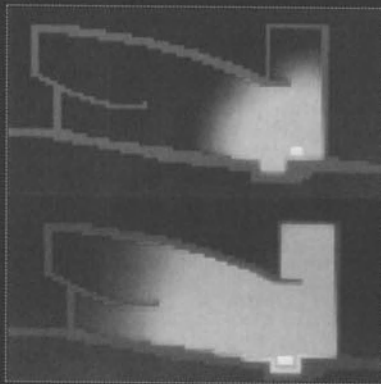
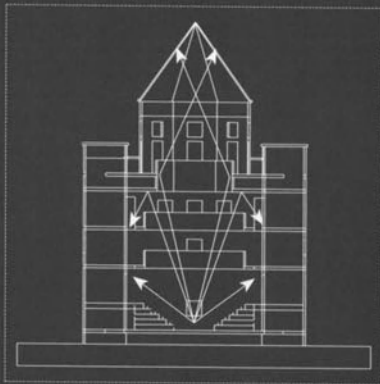
Licht: Simulation

Energie: Abstraktion



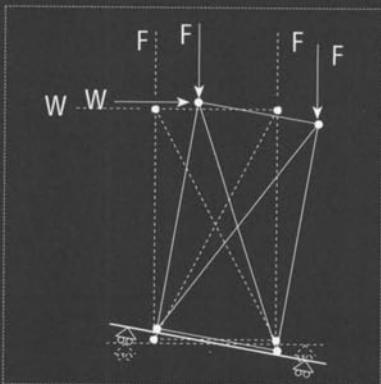
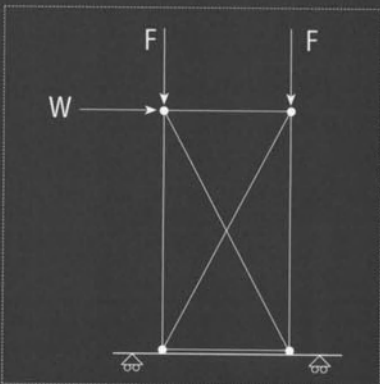
Energie: Simulation

Akustik: Abstraktion



Akustik: Simulation

Statik: Abstraktion



Statik: Simulation

Van der Mark, 1993

Methode: Top-Down

Anwendung: Bauplanung, Softwareplanung, Hierarchische Design-Systeme

Verwandte Gebiete: Staatliche Organisationen

Ausgehend von einer festen Zielvorstellung, dient die Top-Down-Methode der Erzeugung einer Gesamtlösung durch Zerlegung in Teilprobleme. Sie benötigt zu Beginn eine abstrakte und wohldefinierte Problemspezifikation, die danach solange in kleinere Teilspezifikationen zerlegt werden kann, bis primitive Operatoren darauf anwendbar sind [Simon 1992]. Die Methode ist besonders abhängig von der Identifikation passender Abstraktionen, von der Problem-Dekomposition und von der Handhabung und der Lösung von Konflikten, die während des Designprozesses auftreten können. Computerimplementationen benutzen dafür häufig iterative Schritte. Eine Vereinfachung erreicht man, wenn der gesamte Prozeß in eine vorkompilierte Struktur standardisierter Elemente paßt, für die dann lediglich die entsprechenden Parameter zu bestimmen sind. Beginnt man mit einer Beschreibung der Gesamtfunktion des Objektes, so lassen sich die Einzelkomponenten in Bezug auf ihre Eignung für die Unterstützung der Gesamtfunktion und ihre wechselseitigen Beziehungen testen.

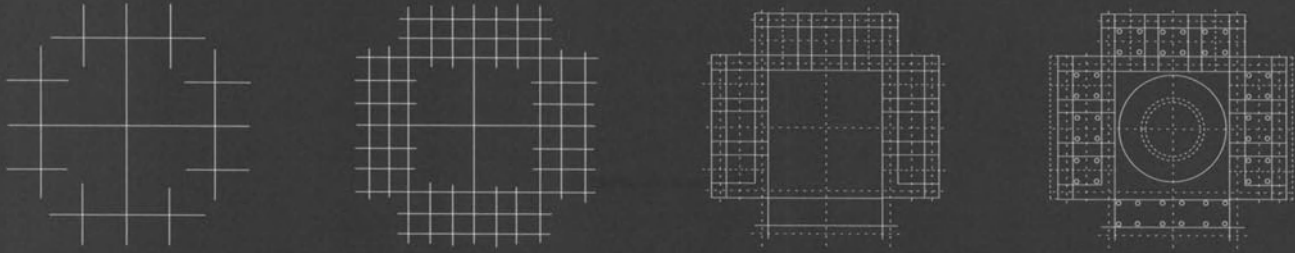
Top-Down, wofür im Deutschen das Wort Verfeinerung keine direkte Übersetzung bietet, ist auch in der Architektur eine etablierte Methode, Probleme zu lösen. Top-Down ist am ehesten mit hierarchischer Komposition oder Dekomposition zu vergleichen. Erfahrene Architekten und Künstler arbeiten mit der Top-Down-Methode: Sie beginnen mit einer schematischen Skizze des Gesamtprojekts und verfeinern diese dann schrittweise, bis der verlangte Grad an Detaillierung und Vollständigkeit erreicht ist.

Die Methode eignet sich bestens für die Rekonstruktion (oder Konstruktion) vergangener Architektur. Die Anlage eines römischen Kastells zum Beispiel ist in lediglich sechs Schritten denkbar: (1) Entscheidung für den Bau eines Kastells. Die politische oder militäri-

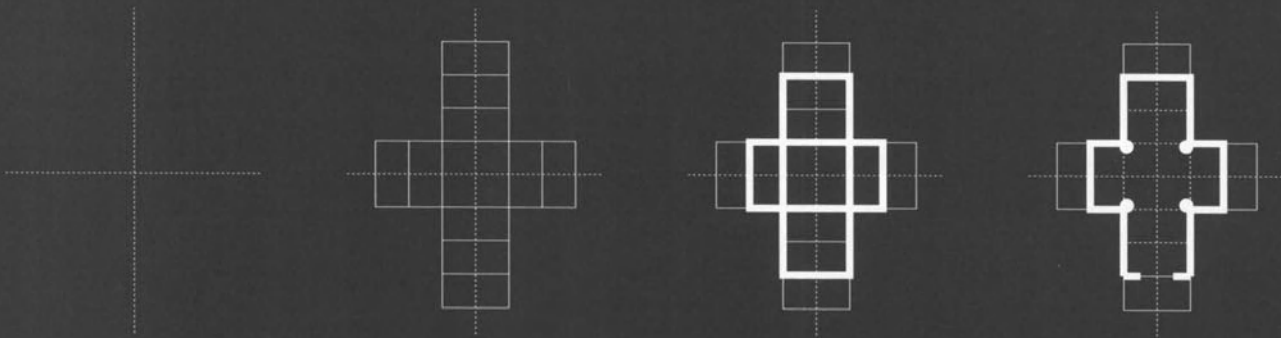
sche Lage lassen ein Kastell notwendig erscheinen. (2) Auswahl des Ortes. Topographie oder militär-strategische Notwendigkeit bestimmen die Lage. (3) Festlegung der Achsen. Nach bekanntem Muster werden Cardo und Decumanus festgelegt. (4) Festlegung der Größe. Die Truppenstärke gibt den Ausschlag. Ist sie bekannt, kann die in vier Quadrate gegliederte Fläche bestimmt werden. (5) Position der Mauern und Tore. Der notwendige Abstand der Truppenunterkünfte von den Mauern ist bekannt. Ebenso eindeutig festgelegt ist damit die Position der Tore. (6) Bau der Einzelelemente. Nach Festlegung der Position und der Größe der Elemente, wie Mauern, Tore, Unterkünfte, Repräsentationsbauten und Straßen, können diese gesondert vergeben werden. Für alle Elemente bestehen bereits Prototypen. Bei der Adaptierung dieser Prototypen gehen die Baumeister vor wie bei der Anlegung des gesamten Kastells.

Der wirkliche Ablauf des Entwerfens und Bauens eines römischen Kastells war möglicherweise weniger direkt und weniger einfach, doch ist es nachträglich möglich, einen solchen Bautyp auf diese Art zu rekonstruieren. Die Beschreibungen von Entwurfsprozessen in den klassischen Traktaten von Vitruv, Palladio und Alberti erinnern ebenfalls stark an ein Top-Down-Vorgehen [Stenvert 1992]. Und gewiß laufen Entwurf und Ausführung vieler Standardgebäude auch heute noch nach diesem Schema ab.

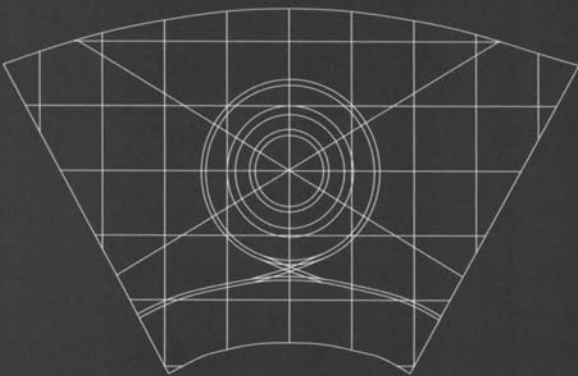
Die Top-Down-Methode eignet sich geradezu ideal für Computerimplementierungen. Sie gewährt Sicherheit im Vorgehen, denn ein Entwurfsstadium baut logisch auf dem vorhergehenden auf. Da eine klare Zielvorstellung existiert, ist der gesamte Prozeß nachvollziehbar. Dies hat Mitchell dazu veranlaßt, ein Computerprogramm zu schreiben, das diese Vorgehensart unterstützt [Mitchell 1990b, S. 137].



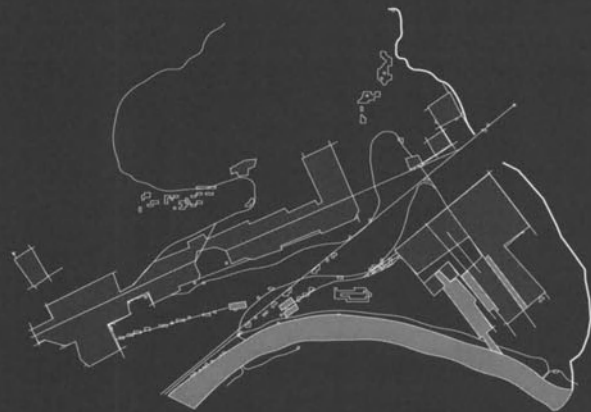
Schrittweise Verfeinerung eines Grundrisses nach Durand



Top-Down Entwicklung eines Grundrisses, Dave, 1993



Schematisches Diagramm der Gartenstadt,
nach E. Howard, 1898



Schematisches Diagramm der Cité Industrielle,
nach T. Garnier, 1904-17

Methode: Bottom-Up

Anwendung: Interaktive, stufenweise Entwicklung von Design, Experiment

Verwandte Gebiete: Formengrammatiken, Produktionssysteme

Die Bottom-Up-Methode ist die Zusammensetzung einer Gesamtlösung durch schrittweise erfolgende, iterative oder rekursive Kombination von Einzelementen. Sie ist nur dann einsetzbar, wenn das Entwurfsproblem durch rationale Entscheidungen gelöst werden kann. Die Einzelemente müssen bekannt und kombinierbar sein, um neue zusammengesetzte Objekte oder Hypothesen erstellen zu können. Auf jeder Stufe des Kombinations- und Generierungsvorganges wird die neu erzeugte Information darauf geprüft, ob sie die anfangs definierte Zielvorstellung erfüllt. Die Methode ist das Gegenstück zur Top-Down-Methode. Die Bottom-Up-Methode verwendet Suchtechniken (Search), um aus verschiedenen Einzelementen Gesamtkompositionen zu generieren. Architekturforscher glauben nachweisen zu können, daß diese Methode bei unerfahrenen Architekten mehr zur Anwendung kommt als bei erfahrenen, die mehrheitlich mit komplexeren Strukturen, wie Prototypen oder mit der Top-Down-Methode, operieren [Akin 1986]. Die Verwendung der Bottom-Up-Methode verlangt auf jeden Fall Wissen über das formale und funktionale Architekturvokabular, um aus Einzelementen eine Gesamtlösung zu komponieren.

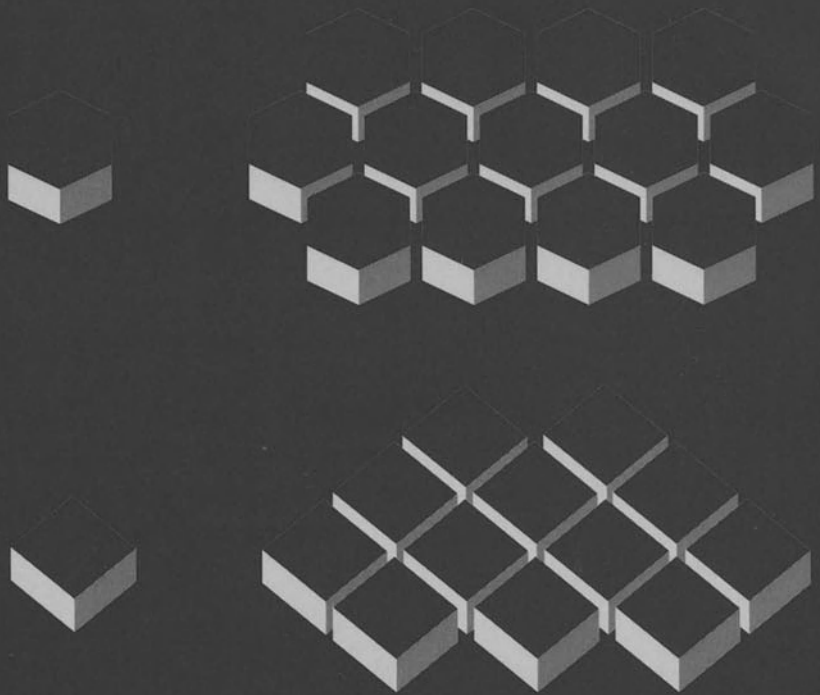
Die architektonische Komposition volumetrischer Grundformen ist ein beliebtes Thema im frühen Entwurfsunterricht und im Werk berühmter Architekten [Le Corbusier 1963]. Das Arbeiten mit Kuben, Sphären, Kegeln, Keilen und deren Kombination ist jedem Architekten bekannt. Niemand wird behaupten, daß sich diese Grundformen direkt aus menschlichen Bedürfnissen ableiten; vielmehr sind es geometrisch-mathematische Konstrukte, die durch das Vorhandensein fundierter Grundaxiome gut kombinierbar sind. Und doch beeinflussen sie die Architektur von der Antike bis zur Gegenwart. Es ist natürlich, daß für die Art der Kombination der Teile Regeln entwickelt wurden. Diese Regeln kön-

nen als Beginn einer formalisierten Bottom-Up-Methode gesehen werden.

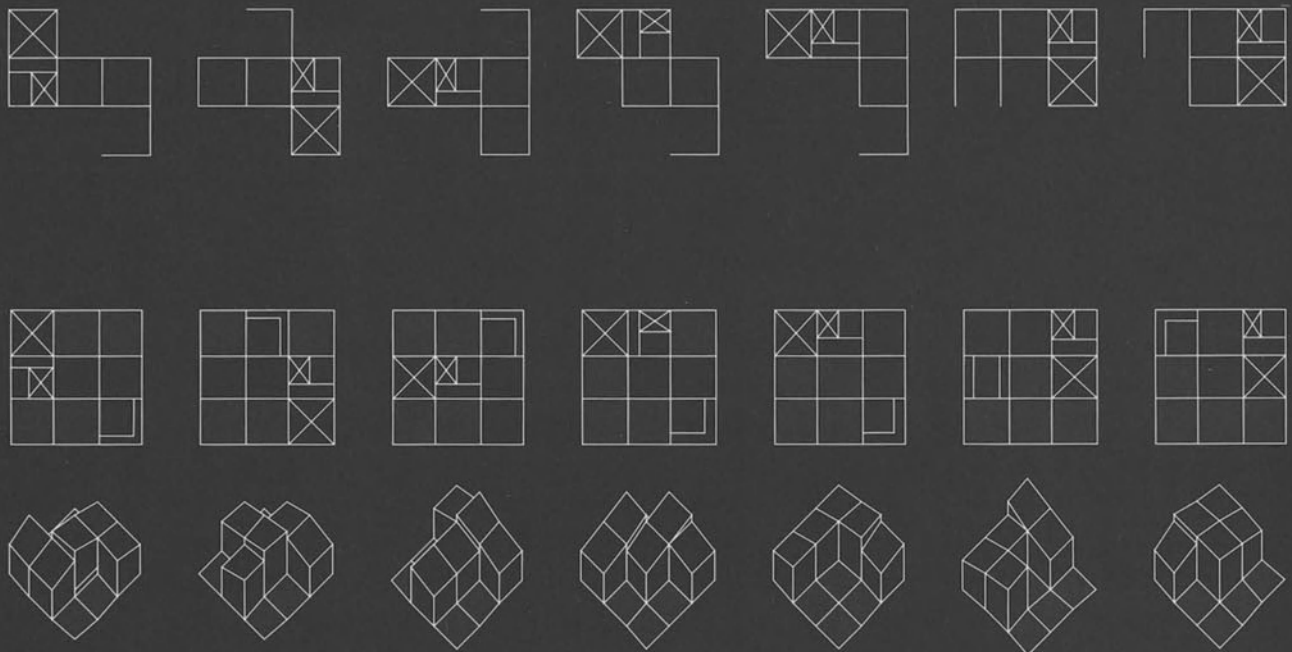
Vereinfacht dargestellt, versucht die Methode durch die Kombination von Einzelementen nach definierten Regeln zu einer architektonischen Gesamtlösung zu kommen. Die Gemeinsamkeit mit der Top-Down-Methode besteht darin, daß das Ziel bekannt ist und dabei hilft, den Prozeß zu steuern. Die entwerferische Freiheit bei der Verwendung der Bottom-Up-Methode scheint größer als die im Top-Down-Verfahren. Damit ist sie prädestiniert für die Exploration neuer Entwurfsideen, für Experimente formaler und inhaltlicher Art. Sie ist allerdings weniger effektiv, wenn es um die schnelle Lösung wohlbekannter Probleme geht. Die Bottom-Up-Methode birgt zudem die Gefahr, immer wieder das Rad neu zu erfinden. Denn wenn sie allein eingesetzt wird, um ein bestimmtes Ergebnis zu erreichen, sind die meisten Zwischenschritte überflüssig. Dagegen bietet sie auch die Chance, bei anscheinend klar definierten Problemen durch kleine Änderungen im Vorgehen oder in den verwendeten Regeln Innovationen zu erzeugen.

Ein typisches Instrument zur Unterstützung dieser Methode sind wissensbasierte Systeme (Produktionssysteme) und deren graphisches Äquivalent, die Formengrammatiken (Shape Grammars). Ausgiebige Studien dieser Methode und ihrer möglichen Verwendung in der Architektur fanden in den frühen achtziger Jahren statt, unter anderem durch Koning und Eizenberg [Koning 1981]. Computertechniken für diese Methode sind Suchstrategien wie Generieren und Testen (Generate and Test) und Maximumsuche (Hill Climbing). Die Bottom-Up-Methode hat mit dem Instrument Shape Grammars den Einzug in den Architekturunterricht erreicht, wie Experimente an der Architekturschule der University of California in Los Angeles [Knight 1992] und an der ETH Zürich zeigen [Schmitt 1989c].

Intuitives Experimentieren oder Anordnung
von Einzelementen nach einfachen Regeln
kennzeichnet die Bottom-Up-Methode.
Dave, 1993



Formalisierung der Regeln und
Übergabe an den Computer. Führt das
Programm eine blinde Suche durch, ist
die Zahl möglicher Lösungen zu groß.
Regeln Kramel, Programm Chen 1991



Methode: Prototypen

Anwendung: Lösung bekannter Probleme, Wissensbasierte Systeme

Verwandte Gebiete: Case-Based Reasoning

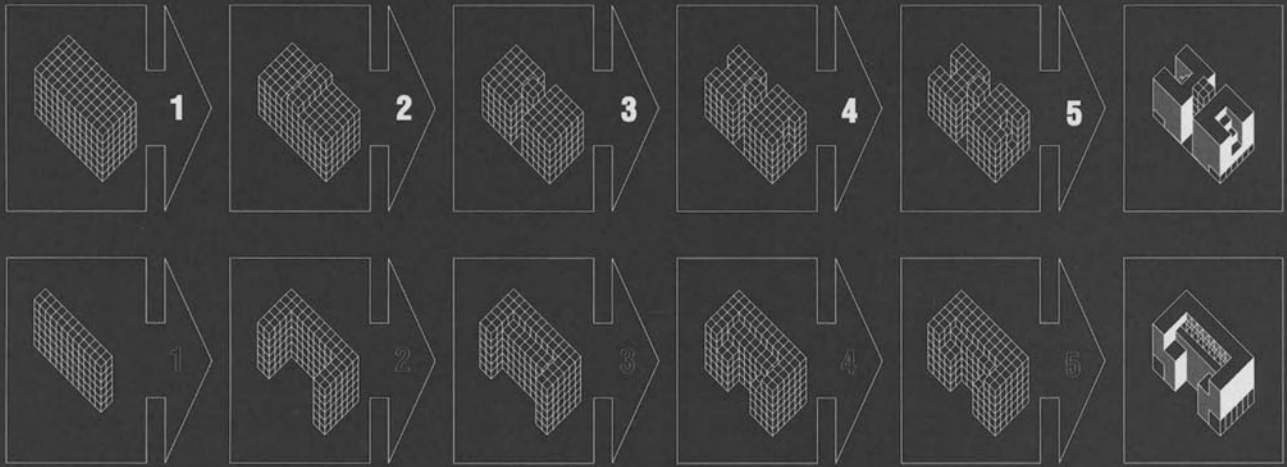
Prototypen sind konzeptionelle Schemata für die Repräsentation generalisierten Entwurfswissens, die hierarchisch vom Allgemeinen zum Spezifischen organisiert sind. Entwerfen mit Prototypen bedeutet die Erzeugung einer Variation eines bestehenden Prototyps, die den Anforderungen des Entwurfs entspricht. Operationen mit Prototypen fallen in drei Klassen: Verfeinerung von Prototypen, Anpassung von Prototypen und Erzeugung neuer Prototypen. Diese Klassifizierung entspricht Anwendungen im routine, im innovativen und im kreativen Design [Gero 1988, Gero 1990, Gero 1993].

Prototyp-Verfeinerung (Prototype Refinement) ist die bisher am besten verstandene und am häufigsten eingesetzte Methode, bei der ein Prototyp variiert wird. Er erfüllt die Anforderungen nur dann, wenn die Entscheidungsvariablen des Entwurfsproblems innerhalb der Grenzen der Prototypvariablen liegen. Das Endprodukt entsteht, indem die Variablen des Prototyps angepaßt werden, ohne den Prototyp grundlegend zu verändern. Die Prototyp-Anpassung (Prototype Adaptation) versucht, eine bestehende Entwurfslösung mit wenigen Modifikationen auf das neue Problem anzuwenden. Dies bedingt die Modifikation des bestehenden Prototyps durch die Einführung neuer Werte für existierende Variablen, neuer Designelemente und neuer Designoperatoren.

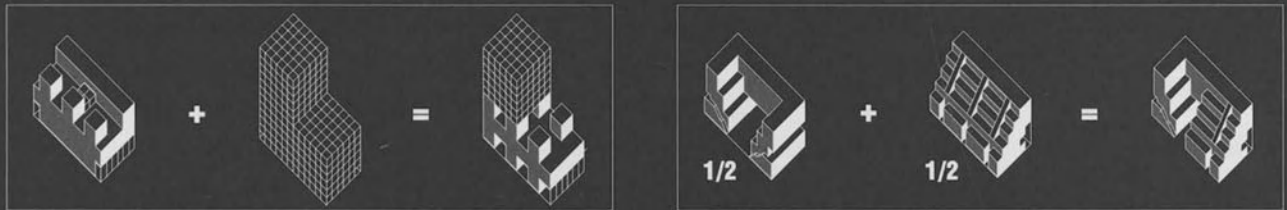
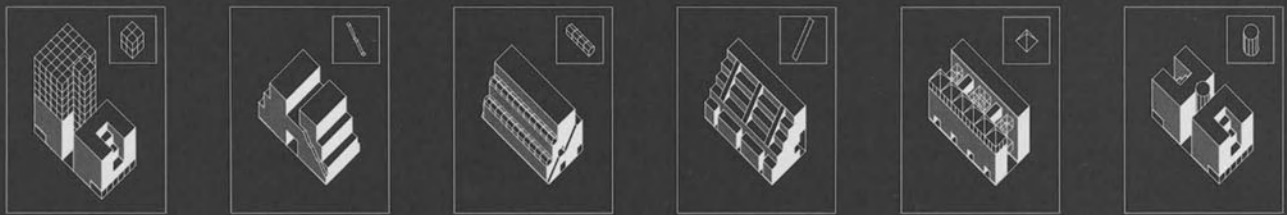
Die Erzeugung neuer Prototypen (Prototype Creation) ist die schwierigste Aufgabe und entspricht einem vollkommen neuen Entwurf. Untersucht wurden bisher in diesem Zusammenhang Konzeptformation (Concept Formation), die sich auf die Variablen konzentriert, sowie Konzeptgruppierung (Concept Clustering), die sich mit der Hierarchie der Attribute beschäftigt. Bekannt sind Prototypen aus der Flugzeug- und Automobilindustrie, also aus einer 'many-of-a-kind'-Produktion. Dort ist die Prototyp-Verfeinerung eine notwendi-

ge Methode, denn im Prototyp entdeckte Fehler können so in der Serienproduktion vermieden werden. In der Architektur, die sich durch eine 'one-of-a-kind'-Herstellung auszeichnet, entwickeln sich Prototypen auf umgekehrtem Weg. Der Prozeß ist beinahe mit der darwinistischen Selektion zu vergleichen; allerdings ist das Testobjekt nicht ein unbewohntes Objekt, sondern eine große Zahl benutzter Gebäude. Bestimmte Gebäudetypen, Materialien oder Konfigurationen bewähren sich im Laufe der Zeit, ohne daß kausale Zusammenhänge bekannt sein müssen. Das Essentielle eines solchen Objektes, sei es ein Möbel oder ein Gebäude, läßt sich dann als Prototyp beschreiben. So gibt es zu jeder bestimmten Zeit das prototypische Einfamilienhaus, das prototypische Bürogebäude oder die prototypische Bank. Zu einem bestimmten Zeitpunkt in der Architekturentwicklung hat jeder Prototyp seine Berechtigung, und ohne viel geistigen Aufwand läßt sich eine akzeptable Lösung durch Anpassung des Prototyps an eine neue Situation erzeugen.

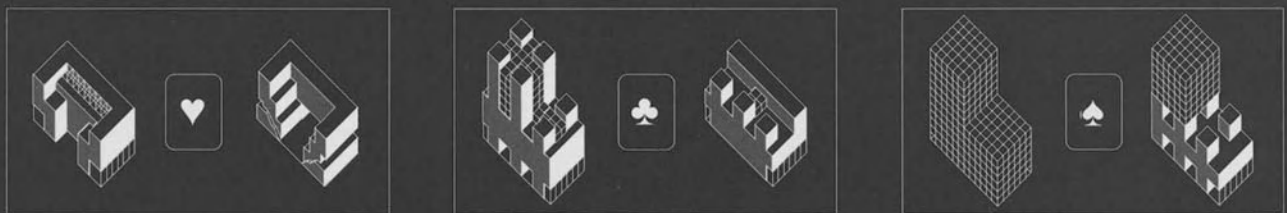
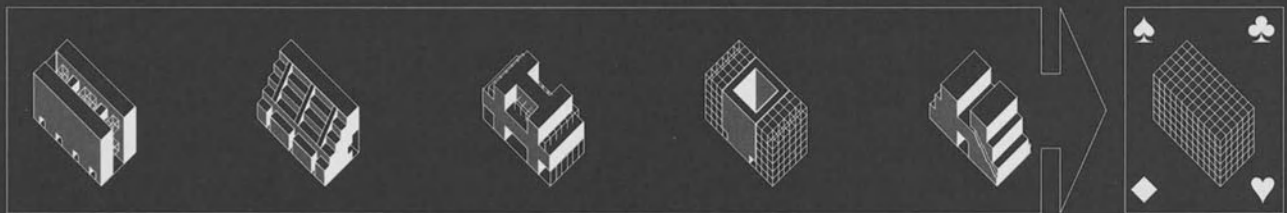
Die Arbeit mit Architekturprototypen ist wesentlich konkreter als die Top-Down- oder Bottom-Up-Methoden, sie ist jedoch beschränkter im Resultat. Es ist möglich, mit Prototypen auf verschiedenen Abstraktionsniveaus zu arbeiten. Die Anpassung eines Prototyps auf das Niveau der Gebäudefunktionen kann auf diese Weise wesentlich größere architektonische Konsequenzen haben als diejenige auf geometrischem Niveau. Das bedeutet, daß ein Gebäudefunktionsprototyp, zum Beispiel in Form eines Diagramms, ebenso anpaßbar ist wie ein geometrischer Prototyp. Im Unterschied zu den Top-Down- oder Bottom-Up-Methoden bedeutet die Arbeit mit einem Prototyp das Verständnis und die Manipulation eines vollständigen Objekts, dessen Teile untereinander klare Beziehungen aufweisen und das als ein Ganzes definiert ist.



Verfeinern von Prototypen



Anpassen von Prototypen



Erzeugen von neuen Prototypen, Chen, 1991

Methode: Fallbasiertes Schließen

Anwendungen: Klassifikation, Evaluation durch Vergleich, Entwurf und Planung

Verwandte Gebiete: Das Phänomen des Sich-Erinnerns

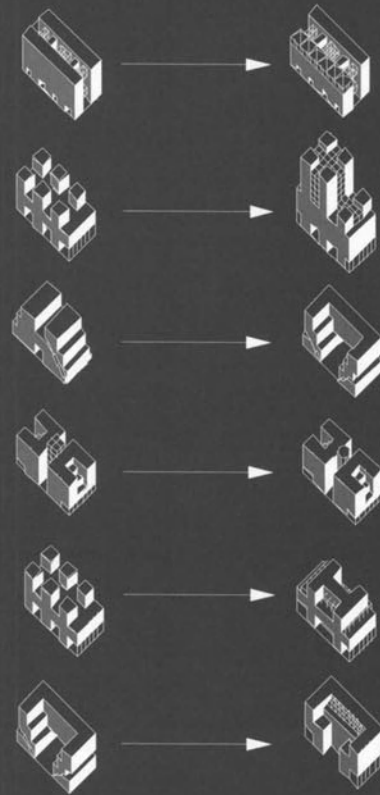
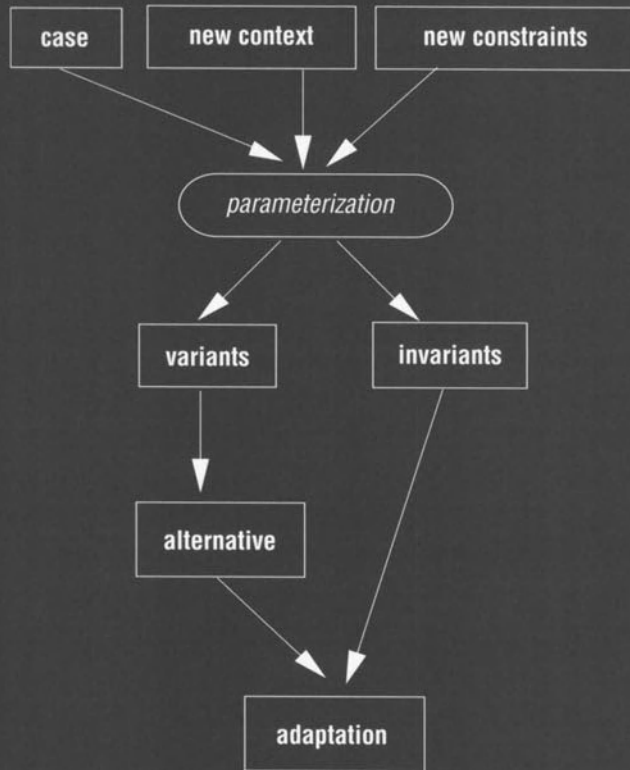
Fallbasiertes Schließen (Case-based Reasoning) ist der Versuch, das Gedächtnis von Experten oder ein reales Objekt in seiner Gesamtheit und unkompiliert zu simulieren. Dadurch unterscheidet es sich von anderen wissensbasierten Systemen, die Expertenwissen vor der Anwendung auf neue Probleme, zum Beispiel in Form von Regeln, kompilieren. Fallbasiertes Schließen benutzt die Funktion des Sich-Erinnerns, um ähnliche Lösungen in der Vergangenheit zu finden [Schank 1982]. Anwendungen des seit Beginn der achtziger Jahre aktiven Forschungsgebietes reichen von der Planung [Hammond 1989] über militärische Anwendungen bis hin zur Architektur. Ein fallbasiertes System geht neue Probleme an, indem es zunächst den am nächsten verwandten Fall sucht und ihn an die neue Situation anpaßt. Zu diesem Zweck sind die Fälle nach verschiedenen Gesichtspunkten indiziert. Ist eine Adaptation nicht möglich, kombiniert und adaptiert das System Teilproblemlösungen mehrerer Fälle. Dies vermeidet die wiederholte Lösung ähnlicher Probleme sowie die Schwierigkeiten in der Formulierung allgemeingültiger Regeln und deren Modifikation für Spezialfälle. Fallbasiertes Schließen bietet auch die Möglichkeit, die eigene Fähigkeit durch die Lern- und Erinnerungskapazität der Maschine zu verbessern.

Fallbasiertes Schließen ist eine in der Architektur altbekannte Methode. Sie besteht darin, für die Lösung eines neuen Problems eine ähnliche Problemstellung und die entsprechende architektonische Lösung in der Vergangenheit zu finden und diese anzupassen. Die Motivation für ein solches Vorgehen ist einfach nachvollziehbar: Einmal gemachte gute Erfahrungen sollen beibehalten, einmal gemachte Fehler nicht wiederholt werden. Die Hauptprobleme sind: die Vollständigkeit der Fall-Datenbank, Methoden der Auffindung korrekter Fälle und die Anpassung an die neuen Anforderungen sowohl in geo-

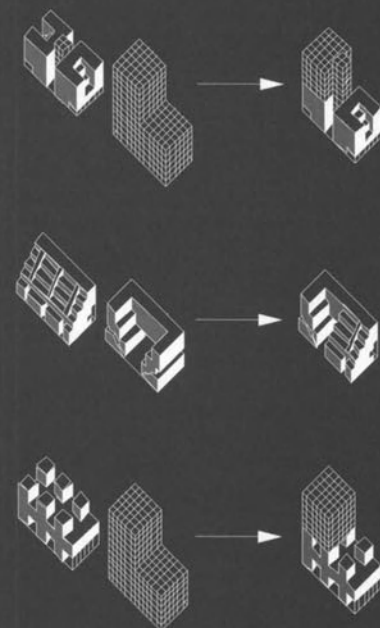
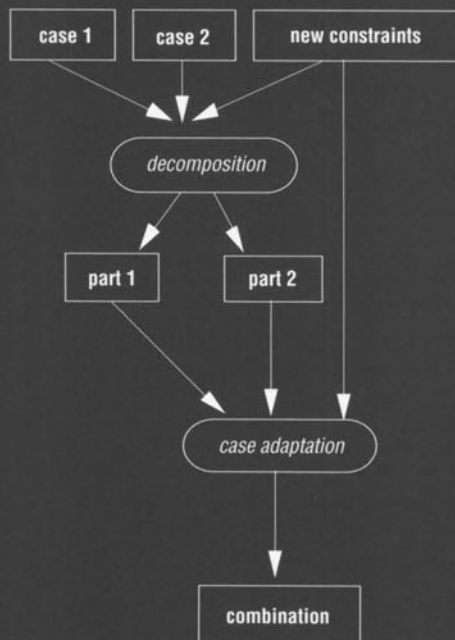
metrischer als auch in topologischer Hinsicht. Die größte und kompletteste Fall-Datenbank ist die gebaute Umwelt, die wir täglich erleben und die als Basis für alle neuen Entwürfe dient. Eine erste Stufe der Abstraktion sind die beschreibenden Werke der Architekturgeschichte, die verschiedene, nicht aber alle Aspekte eines Gebäudes erläutern und einen gewissen Anspruch auf Ausgewogenheit erheben. Eine weitere Abstraktionsstufe sind Architekturzeitschriften, die ausgewählte Bauten partiell beschreiben. Zu Beginn des Architekturstudiums besteht die eigene Fall-Datenbank aus den unmittelbaren Erfahrungen der umgebenden Architektur. Im Verlauf der Ausbildung wächst die Zahl der bekannten Fälle ständig an. Die wichtigen Erfahrungen mit gebauter Architektur entwickeln sich mit den ersten eigenen Projekten. Erst gegen Ende einer Architektenkarriere ist die Fall-Datenbank von überzeugender Größe.

Der Anpassungsprozeß oder die Adaptation bestehender Architektur auf neue Probleme ist ein komplexer Vorgang. Die einfachste Stufe ist die Null-Adaptation oder die direkte Übernahme. Bei der Nutzung der eigenen Fall-Datenbank ist dies möglich, bei der Verwendung von Fällen anderer Architekten können daraus rechtliche Probleme entstehen. Bei Architekturstudenten ist, von Ausnahmen abgesehen, die Tendenz zum Plagiat zum Glück gering. In der nächsthöheren Stufe werden Teile von Architekturlösungen übernommen, andere geometrisch oder nach Materialien angepaßt. In der kompliziertesten, aber auch interessantesten Form der Adaptation werden topologische Änderungen vorgenommen. Am Ende jeder Adaptation steht die Evaluation. Sie überprüft, ob die Wahl und die Anpassung des ausgewählten Falles das gewünschte Resultat erbringen. Das Ergebnis der Evaluation ist die endgültige Akzeptanz des neuen Objekts oder der Wiedereinstieg in eine frühere Phase des Projekts.

Anpassen von Fällen -
Case Adaptatio., Shih, 1993



Kombination von Fällen -
Case Combination, Shih, 1993



Methode: Maschinenlernen

Anwendung: Alle Bereiche des intelligenten CAAD

Verwandte Gebiete: Kognitive Psychologie

Maschinenlernen (Machine Learning) hat zum Ziel, durch automatische Wissensaneignung und Wissensverfeinerung die Qualität von Programmen zu verbessern. Ein Machine Learning-System nimmt Beobachtungen auf und verarbeitet sie in einer Wissensbasis. Nach Michalski fällt Machine Learning in drei logische Kategorien [Michalski 1986]:

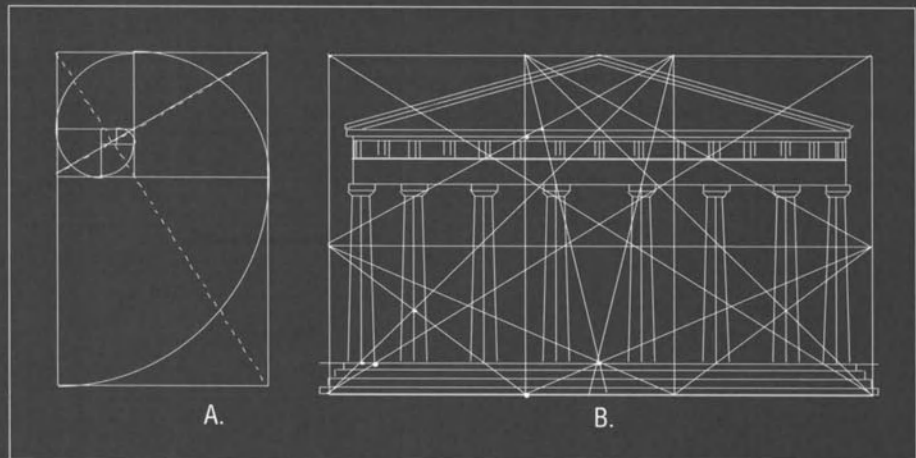
(1) Lernen durch Deduktion. Dabei leistet ein Programm deduktive Inferenzen und reformuliert die Ableitungen in Form nützlicher Schlüsse. Die ursprüngliche Aussage wird beibehalten. Erklärungs-basiertes Lernen ist ein typisches Beispiel. (2) Lernen durch Analogie. Es besteht aus dem unveränderten Übertragen von Wissen aus vergangenen Problemlösungen auf ein neues Problem, das vieles mit den vorherigen gemeinsam hat, sowie aus der Anwendung und Anpassung des übertragenen Wissens zur Lösung eines neuen Problems. Dabei besteht zwischen der ursprünglichen Information aus vergangenen Problemlösungen und der neu gewonnenen Information lediglich ein loser Zusammenhang. Man könnte sagen, daß die ursprüngliche Aussage zwar nicht beibehalten wird, die Plausibilität zwischen bestehender und neuer Problemlösung aber gewahrt bleibt. (3) Lernen durch Induktion. Allgemein als der Vorgang des Entdeckens durch die Analyse und Kombination bestimmter Einzelheiten bezeichnet, besteht Lernen durch Induktion aus dem Ziehen von Schlüssen aus einer Reihe von Beobachtungen. Dabei besteht zwischen den ursprünglichen Phänomenen und den Entdeckungen meist kein direkter Zusammenhang, so daß man in diesem Fall lediglich die Wahrung der Unterschiede zwischen altem und neuem Wissen beobachten kann.

Lernen ist eine Grundvoraussetzung für das Schaffen von Architektur, durch Jahrhunderte bewährt im Meister-Schüler-Verhältnis sowie durch das Betrachten,

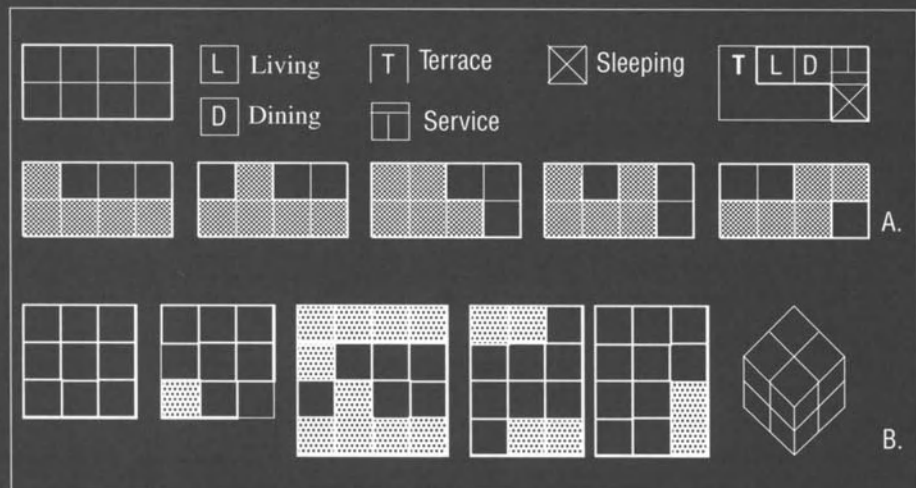
Zeichnen und Verstehen von gebauter Architektur. Von den bereits entdeckten Lernmechanismen sind für die Architektur besonders das Lernen durch Deduktion, Lernen durch Analogie und Lernen durch Induktion interessant. Analoges Lernen spielt eine hervorragende Rolle im kognitiven und besonders im kreativen Bereich [Keane 1988]. Nach Wolstencroft gibt es dabei sieben Stufen [Wolstencroft 1989]: (1) die Erkenntnis, daß Analogie zu einer Lösung des Problems führen kann, (2) das Finden einer ähnlichen Situation in der Vergangenheit, (3) deren Aufarbeitung in Hinblick auf das neue Problem, (4) das In-Beziehung-Setzen (Mapping) entsprechender Problemteile, (5) das Finden neuen Wissens, (6) die Rechtfertigung der Analogie und (7) die Konsolidierung der Analogie. Forschungsergebnisse liegen hauptsächlich für das Mapping und das Finden neuen Wissens vor. Evans versuchte bereits zu Beginn der sechziger Jahre eine Computerimplementierung [Evans 1963]. Die heute am weitesten akzeptierte Form der Analogie ist die voraussagende (predictive) Analogie: Haben zwei Situationen gewisse Ähnlichkeiten, so sind weitere Ähnlichkeiten wahrscheinlich; dabei ist eine Situation meist besser bekannt (Source), während die neue Situation das zu lösende Problem (Target) ist. Eine weitere interessante Form der Analogie ist die Metapher, eine Neuinterpretation eines Symbolsystems in einer anderen Umgebung als der, in die das erste System gehört. Laseau nennt wichtige Quellen von Metaphern im Entwurf [Laseau 1980].

Eine wichtige Anwendung des induktiven Maschinenlernens in der Architektur hat Chen entwickelt. Sein Programm generiert eine Anzahl architektonischer Alternativen zu einem Entwurfsproblem. Es folgt die Auswahl von Alternativen nach eigener Präferenz. Beim nächsten Vorschlag berücksichtigt das Programm diese Präferenzen bereits [Chen 1991].

A: Rechtecke im Goldenen Schnitt
B: Induktives Auffinden des Goldenen Schnittes in der Fassade des Parthenon, Chen, 1991



A: Layout-Programm angewendet auf ein 2x4 Raster. Nach der Selektion der optimalen Lösungen werden diese in B auf andere Raster angewendet. Regeln Kramel, Programm Chen, 1991



Induktion von Entwurfsregeln durch das Beobachten und Verarbeiten guter (A) und schlechter (B) Beispiele. Ein Experte auf dem entsprechenden Gebiet wählt die guten und schlechten Beispiele aus; das Programm findet danach die neuen Regeln. Regeln Kramel, Programm Chen, 1991

A. Gute Beispiele:



B. Schlechte Beispiele:



Instrumente für den computerunterstützten Entwurf

Instrumente entwickeln sich meist aus einem konkreten Bedürfnis und werden auf eine bestimmte Anwendung zugeschnitten. Sie sind deshalb mehr als Methoden an ein Einsatzgebiet gebunden. Diese Beobachtung bestätigt sich vor allem bei den verschiedenen Modellierprogrammen, die von einfachen zweidimensionalen Paint-Systemen bis zu komplexen und integrierten mehrdimensionalen Simulationsprogrammen reichen. Eine große Breite haben auch die verschiedenen Programmierinstrumente erreicht, welche die Verwirklichung von formalisierbaren Vorstellungen erlauben. Computer-Instrumente dienen entweder zum Finden einer ersten möglichen Lösung, verschiedener Lösungen, aller möglichen Lösungen oder einer optimalen Lösung. Von besonderem Interesse bei der Entwicklung intelligenter Entwurfsumgebungen sind die folgenden Instrumente:

Editieren I - Zahlen und Text. Textverarbeitung und Tabellenkalkulation sind die Grundinstrumente heutiger Büroarbeit auch im Bereich Architektur. Sie sind ausgereifte Werkzeuge für Berechnungen, das Beschreiben von Architektur, die Bürokommunikation und das Erstellen von Programmen.

Editieren II - Diagramme. Das traditionelle Entwerfen auf Papier erhält Konkurrenz durch computerbasierte Diagrammsysteme, die eine wesentlich größere Informationstiefe erreichen. Diese Entwicklung befindet sich im Anfangsstadium.

Editieren III - Geometrie. Operationen im dreidimensionalen Euklidischen Raum nach geometrischen Gesetzmäßigkeiten sind heute Teil jedes CAD Systems. Geometrie-Editoren werden heute zunehmend mit anderen Instrumenten integriert.

Programmieren I - Traditionelle Programme. Das Programmieren sequentieller Abläufe ist für viele Teilgebiete des Entwerfens ein wichtiges Hilfsmittel. Laien

können einfache Programme, die immer wiederkehrende Abläufe beschreiben, mit Makrosprachen herstellen.

Programmieren II - Wissensbasierte Programme. Programme dieser Art beinhalten die Sammlung von Fakten, darauf anzuwendende Regeln, sowie Such- und Konfliktlösungsstrategien. Für nicht exakt definierte Probleme und für die Anwendung von Faustregeln setzt man Hoffnungen auf diese Art des Programmierens.

Programmieren III - Objektorientierte Programme. Sie beinhalten die Definition von Objekten und ihrer Eigenschaften sowie die Beschreibung ihres Verhaltens und ihrer Beziehungen untereinander. Lassen sich hierarchische Zusammenhänge klar definieren, so ist diese Art des Programmierens eine effektive Hilfe, reale Zustände im Computer zu simulieren.

Generieren I - Formengrammatiken. Darunter versteht man eine Sammlung von Formen und darauf anzuwendende Transformationsregeln. Sie sind das graphische Äquivalent der Produktionssysteme, einer Unterabteilung der wissensbasierten Programme.

Generieren II - Fraktale. Dies sind Systeme selbstähnlicher Formen, die aus der Natur abgeleitet und rekursiv im Maßstab vergrößert oder verkleinert werden können.

Parametrisierung. Darunter versteht man die Ausnutzung gemeinsamer Topologie von Objekten für die geometrische Anpassung an neue Problemstellungen.

Objekt-orientiertes Modellieren I - Typen und Variationen. Dieses Hilfsmittel erlaubt die Definition wichtiger Elemente als Typen, die dann in wechselnder Form als Varianten (Instances) wiederverwendet werden können.

Objekt-orientiertes Modellieren II - Substitution. Aus verschiedenen Typen entstehen Modelle. In diesen Modellen können Typen gegeneinander substituiert

oder ausgetauscht werden. Damit lassen sich Alternativen studieren.

Objekt-orientiertes Modellieren III - Detaillierungsgrade. Für jeden definierten Typ lassen sich beliebig viele Detaillierungsgrade entwickeln. So kann ein Modell schrittweise verfeinert werden.

Objekt-orientiertes Modellieren IV - Designfokus - Logical Zoom. Interessiert nur ein Teil eines Entwurfs, so erlauben Detaillierungsgrade und Substitution das Fokussieren auf einen ganz bestimmten Ausschnitt. Damit wird ein interaktives Herangehen an die Teile des Entwurfs möglich, die von größtem Interesse sind.

Für mechanisch und manuell hergestellte Objekte gilt, daß die zur Bearbeitung verwendeten Methoden und Instrumente mitbestimmend für die Qualität des Produktes sind. Die These ist hier, daß ähnliches für intel-

lektuelle Methoden und Instrumente gilt: Je besser und intelligenter die Instrumente, desto kompetentere und qualitativ bessere Ergebnisse lassen sich damit bei richtiger Anwendung erzielen. Dabei ist die Annahme, daß es wie im mechanischen Bereich kein Universalwerkzeug gibt, das alles kann, sondern eine Sammlung präziser Instrumente, von denen jedes einzelne jeweils für einen bestimmten Zweck geeignet ist. Dies ist wichtig, denn ein Computer-System ist immer nur so gut wie die Abstimmung auf das Objekt oder den Vorgang, den es unterstützt. Die Instrumente eröffnen der Architektur neue, mit traditionellen Mitteln nicht gegebene Vorgehensweisen. Es folgt die detaillierte Beschreibung der bereits bekannten und computer-kompatiblen neuen Instrumente für den architektonischen Entwurf und die verwandten Gebiete.

Methoden

Abstraktion und Modellbildung
 Simulation
 Top-Down
 Bottom-Up
 Prototyp-Verfeinerung
 Fallbasiertes Schließen
 Maschinelernen

Zusammenstellung bisher bekannter Methoden und Instrumente

Instrumente

Editieren I - Zahlen und Text
 Editieren II - Diagramme
 Editieren III - Geometrie
 Programmieren I - Traditionelle Programme
 Programmieren II - Wissensbasierte Programme
 Programmieren III - Objektorientierte Programme
 Generieren I - Formengrammatiken
 Generieren II - Fraktale
 Parametrisierung
 Objekt-orientiertes Modellieren I - Typen und Variationen
 Objekt-orientiertes Modellieren II - Substitution
 Objekt-orientiertes Modellieren III - Detaillierungsgrade
 Objekt-orientiertes Modellieren IV - Designfokus - Logical Zoom

Instrument: Editieren I - Zahlen und Text

Anwendung: Textverarbeitung, Tabellenkalkulation

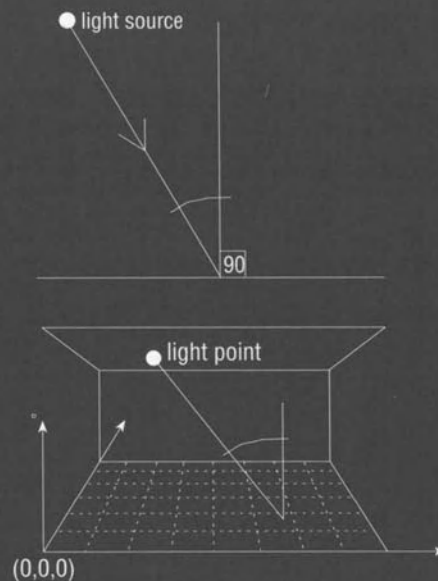
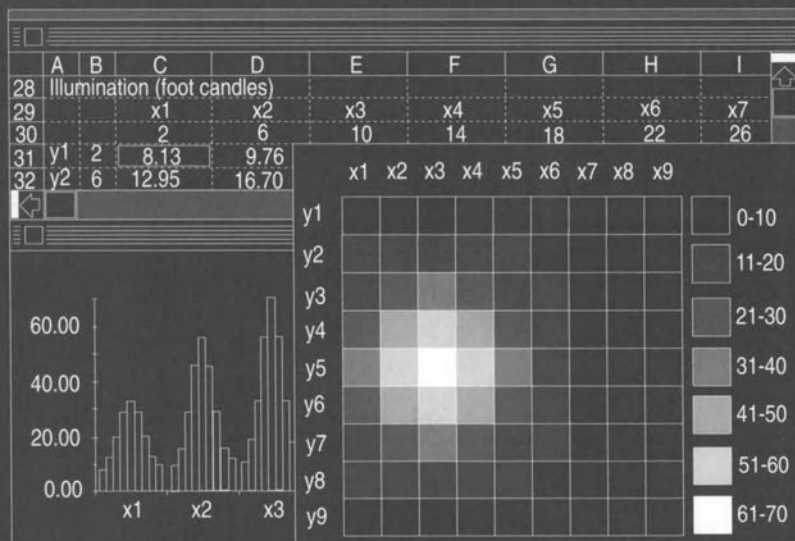
Verwandte Gebiete: Programmieren

Viele Anwendungen des Computers in der Architektur beinhalten die Repräsentation eines Gebäudes in Form von Zahlen oder Text. Zahlen und Text, zusammen mit den darauf ausführbaren Operationen, bilden ein abstraktes Gebäudemodell. Instrumente zur Manipulation dieses Gebäudemodells sind Zahlen- und Texteditoren. In der Entwicklung des architektonischen Entwurfs kommen Zahlen und Text in vielfältigster Weise zur Anwendung: quantitative Evaluation des Gebäudeverhaltens, statische Berechnungen, Beschreibung der Gebäudematerialien, sowie Kostenschätzungen. In einigen Gebieten sind Zahlen und Text die effektivsten Mittel zur Beschreibung und Manipulation von Gebäudeeigenschaften. Um die Idee der Zahlen und Texteditoren zu verdeutlichen, folgt die Beschreibung deren Anwendung in einigen Schritten im Entwurfsprozeß:

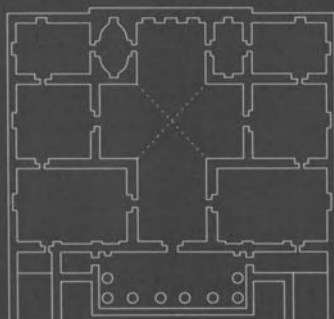
Im frühen Entwurfsstadium, wenn die endgültige Form des Gebäudes noch flexibel ist, werden sinnvollerweise bereits Energie-, Kosten- und Lichtsimulationen durchgeführt, um die Einflüsse dieser Größen auf die Architektur bestmöglich zu erkunden. Im Fall der Energiesimulation liegen die Wetterdaten eines Gebietes für ein ausgewähltes oder ein Durchschnittsjahr in Zahlenform vor. Zahlen repräsentieren das Gebäudevolumen, die Öffnungen, die Materialeigenschaften und alle anderen zur Beschreibung der thermischen Qualität notwendigen Eigenschaften. Die Verarbeitung dieser Zahlen, basierend auf allgemeingültigen Modellen der Mathematik und der Physik, resultiert in der Berechnung von Kenngrößen. Werden die Zahlen und ihre Abhängigkeiten in einem Tabellenkalkulationsprogramm (Spreadsheet) dargestellt, so lassen sich auch schnelle 'was wäre, wenn'-Berechnungen durchführen. Die Drehung des Gebäudes um seine Mittelachse um einige Grad kann Auswirkungen auf den Energieverbrauch haben, ebenso wie viele andere Faktoren, deren wech-

selbige Abhängigkeit durch die Struktur des Modells gewährleistet wird [Schmitt 1983]. Eine weitere Anwendung für Zahlen- und Texteditoren sind Kostenberechnungen, besonders wenn sie nach computerfähigen Methoden durchgeführt werden. Darin sind jedem Element, das durch eine eindeutige Zahlen- und Buchstabenkombination definiert ist, Kostendaten zugeordnet. So lassen sich die Kostengruppen eindeutig identifizieren, und Gebäude lassen sich nach den verschiedensten Kriterien untersuchen.

Im Entwurf können Zahlen Verhältnisse repräsentieren, denen architektonische Bedeutung zugemessen wird [Doczi 1981]. Die Proportionen, auf denen die klassische Architektur aufbaut, aber auch neuere Proportionsysteme, wie Le Corbusiers Modulor, stützen sich auf Zahlen zur Generierung von Entwürfen. In den genannten Fällen ist dies einfach, da eine Affinität zwischen Geometrie, Zahlen und Text besteht: Bei der Erwähnung des Goldenen Schnitts (Text) tauchen sofort die entsprechende Figur (Geometrie) und der Wert 1,618... (Zahl) auf. Ähnliches ist bei der Erwähnung von Wurzel 2 (Text) zu beobachten, wobei die einfache Konstruktion (Geometrie) und der Wert 1,414... (Zahl) assoziiert werden. Viele CAD-Systeme erlauben die einfache Kombination von Text und Zahlen für die Proportionierung von Elementen. Ein weiteres Anwendungsgebiet von Zahlen- und Texteditoren in der Architektur sind in anderen Wissenschaften entwickelte mathematische Techniken. Lineares und dynamisches Programmieren erlauben die Berechnung akzeptabler Lösungen, falls zu Beginn eine Menge von Parametern, eine Menge von Restriktionen (Constraints) und eine Menge von Zielvorgaben (Objectives) vorhanden sind [Radford 1988]. Typische Anwendungsgebiete sind die Optimierung des Energieverbrauchs und das Facility Management.

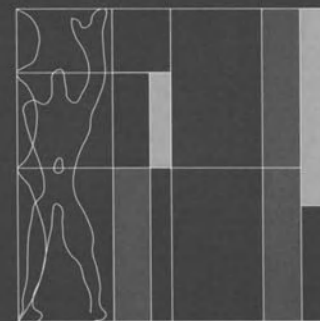


Die Beleuchtung durch eine künstliche Lichtquelle ist abhängig von deren Intensität, dem Einfallswinkel und dem Abstand zum Objekt. Darstellung durch gemischte graphische und numerische Beschreibung in einem Tabellenkalkulationsprogramm. Dave, 1993

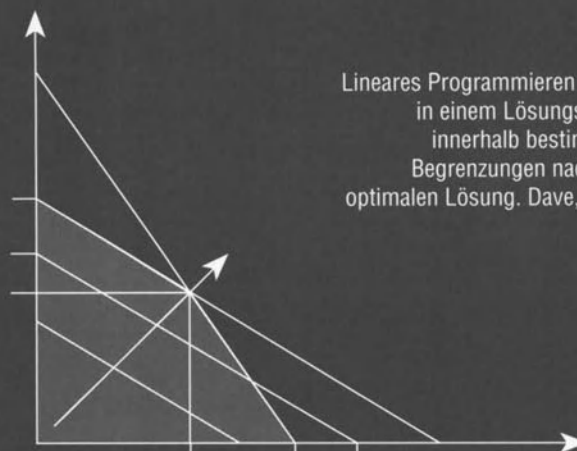
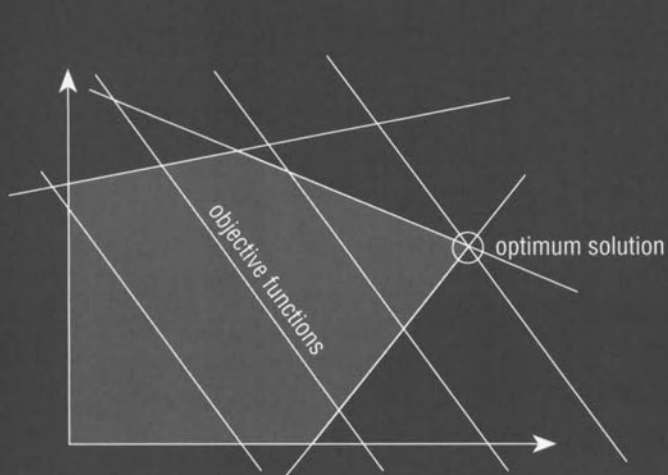


Die Villa Malcontenta von Andrea Palladio und deren zugrundeliegendes Proportionssystem

3:4	2:3	3:4	2:3	3:4
1:1	1:2	1:1	1:2	1:1
2:3	1:1	2:3	3:8	



Das Proportionssystem Modulor von Le Corbusier



Lineares Programmieren sucht in einem Lösungsraum innerhalb bestimmter Begrenzungen nach der optimalen Lösung. Dave, 1993

Instrument: Editieren II - Diagramme

Anwendung: Vorläufiges Schließen, früher Entwurfsprozeß

Verwandte Gebiete: Graphische Repräsentation

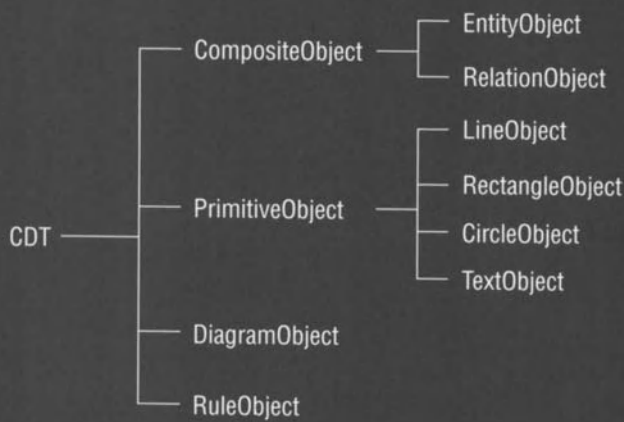
Diagramme sind graphische Repräsentationen symbolischer Zusammenhänge. Die entsprechende Abstraktion ist die funktionale Beschreibung von Objekten, das zugehörige Modell besteht aus der Struktur des Diagramms und den daran auszuführenden Operationen, die im folgenden beschrieben werden. Diagramme lassen vorläufige Schlußfolgerungen und Verbindungen auf einem abstrakten Niveau zu. Sie erlauben es, sich auf bestimmte Aspekte einer Situation zu konzentrieren, ohne den gesamten Problembereich erfassen zu müssen. Oft ist es am einfachsten, komplexe Sachverhalte mit einem Diagramm darzustellen [Larkin 1987]. Diagramme werden meist schnell auf Papier aufgezeichnet und danach in andere Repräsentationen übersetzt. Computerbasierte Diagramm-Editoren sind selten. Bisher war es notwendig, in einer genau definierten Weise eine Textrepräsentation zu erstellen, aus der sich ein Diagramm automatisch ergab. Dies ist jedoch nicht die Art, in der wir mit Diagrammen zu arbeiten gewohnt sind. Vielmehr sollte der Prozeß umgekehrt verlaufen: Aus einem Diagramm ist eine Textdatei zu entwickeln, die später für andere Zwecke weiterverwendbar ist. Eines der ersten Programme, das diese Zielvorstellung realisiert, ist CDT [Dave 1993].

CDT hat zum Ziel, abschnittsweise die Repräsentation und Exploration von Information zu unterstützen. Dies basiert auf zwei grundlegenden Annahmen: (1) Diagramme bestehen aus Objekten und Beziehungen zwischen diesen Objekten, (2) Diagramm-Darstellungen bestehen aus graphischen Symbolen, die im Raum angeordnet sind und durch topologische und geometrische Beziehungen problemrelevante Information darstellen. CDT realisiert diese Bedürfnisse durch die Verwirklichung verschiedener Ideen aus der Computerwissenschaft: Graphische Benutzeroberfläche, Darstellungen von Restriktionen (Constraints) durch Demon-

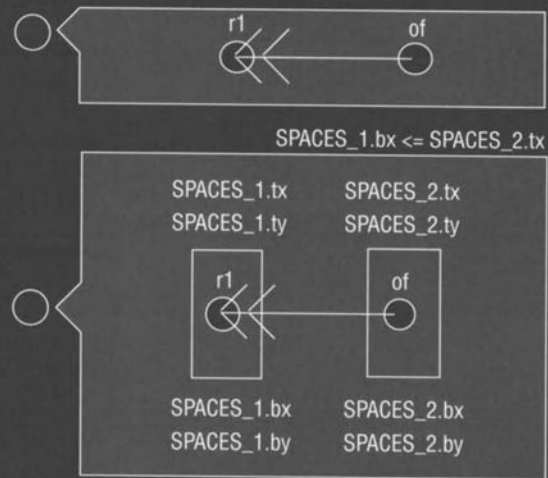
stration sowie Spezifikation und Einhaltung graphischer Kompositionsregeln.

Diagrammatische Anmerkungen sind Regeln, die auf ein Alphabet graphischer Zeichen angewendet werden. Eine Menge solcher Zeichen kann durch graphisches Zusammenstellen - unter Einhaltung von räumlichen Beziehungen - zur Entwicklung eines Formenvokabulars verwendet werden. So entstandene Formen eines Vokabulars können danach dazu verwendet werden, weitere Formen nach einer festgelegten Syntax herzustellen. Diese zusammengesetzten Formen (Compound Forms) können wie Sätze (Expressions) in einer Diagrammsprache verwendet werden. Ein solcher Satz erhält erst dann Bedeutung, wenn jede Vokabel des Satzes mit Information belegt ist, die für das betreffende Anwendungsgebiet von Bedeutung ist. Aufbauend auf diesen Ideen, kann eine diagrammatische Darstellung auf folgenden Komponenten in CDT aufgebaut werden: ein finites Alphabet von primitiven Symbolen, ein möglicherweise infinites Lexikon, Syntaxregeln zur Definition von Assoziationen und Verbindungen, sowie eine Menge semantischer Interpretationen.

CDT ist ein einfacher Diagramm-Editor, der nicht an bestimmte Diagramm-Konventionen gebunden ist. Vielmehr stellt CDT alle notwendigen Werkzeuge zur Definition solcher Konventionen bereit. Ein Anwender definiert ein Diagramm in CDT, (1) indem er zunächst problemspezifische graphische Symbole definiert oder abrufen, (2) diese Symbole mit zugehörigem Textattributen versieht und schließlich (3) die Kompositionsregeln graphisch definiert. Das so hergestellte Diagramm ist später vielfach einsetzbar, indem der Anwender sich auf das Problem konzentriert und CDT die Konsistenz der Beziehungen und der syntaktischen Regeln garantiert und unterhält. Ein so hergestelltes Diagramm ist eine computer-kompatible Beschreibung eines Problems.



Objektklassen in CDT, einem computerbasierten Diagramm-Editor. Dave, 1993



Graphische Herstellung von Beziehungen zwischen Objekten im Diagramm. Dave, 1993

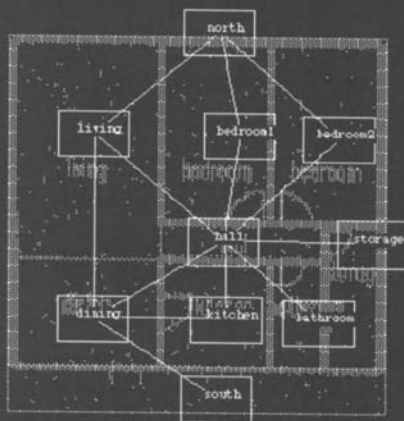


Diagramm mit importierten Geometrie- und Bezugsdaten. Dave, 1993

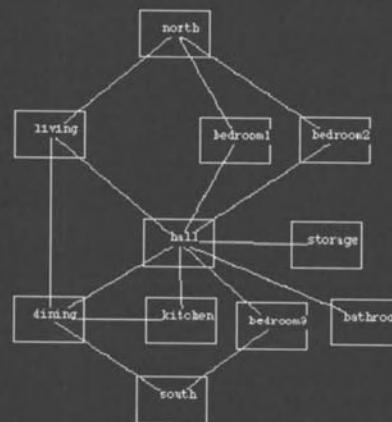
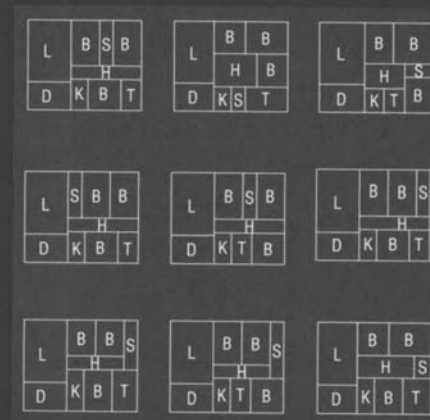


Diagramm der notwendigen Raumbezüge. Dave, 1993

```

:site 1000 800
:spaces 13
;; FUNCTION xmin xmax ymin ymax xmin
1 north 1000 1000 0 0 -1
2 south 1000 1000 0 0 -1
3 east 0 0 800 800 -1
4 west 0 0 800 800 -1
5 hall 120 800 120 800 -1
6 living 350 700 350 700 -1
7 dining 250 500 250 500 -1
8 kitchen 150 500 150 500 -1
9 storage 120 500 120 500 30000
10 bedroom3 250 550 250 550 80000
11 bedroom 180 400 180 400 -1
12 bedroom1 250 550 250 550 80000
13 bedroom2 250 550 250 550 80000
:dependencies 1
14 living hall 100 north
    living dining 150 dining
    hall kitchen 100 dining
    hall storage 100 north
    all bedroom3 100 hall
    
```

Exportierte Daten aus dem Diagramm-Editor. Dave, 1993



Aus den exportierten Daten generierte Grundriss-Varianten. Shih, 1993

Instrument: Editieren III - Geometrie

Anwendungen: Paint-Systeme, Image Processing, CAD

Verwandte Gebiete: Darstellung und Verarbeitung graphischer Information

Geometrie-Editoren sind der Hauptbestandteil kommerzieller CAD-Programme. Grundlage der Geometrie-Editoren ist das geometrische Modell sowie alle darauf anwendbaren Operationen wie Transformationen. Die bekanntesten Editoren sind als Instrumente für die Fläche und für den Raum konzipiert. Sie erlauben die Manipulation von Pixels, Vektoren, Flächen und Volumen. Ein Pixel definiert Punkte auf dem zweidimensionalen Computerbildschirm. Vektoren werden durch Anfangs- und Endpunkt definiert und sind unter anderem in Länge, Position, Lage und Strichstärke manipulierbar. Linien definieren die Ränder von Flächen, Flächen begrenzen dreidimensionale Objekte. Diese Objekte sind als Drahtmodelle (Wireframes), Oberflächenmodelle (Surface Models) und Volumenmodelle (Solid Models) darstellbar.

Drahtmodelle sind durch Eckpunkte und die sie verbindenden Kanten definiert. Der Name leitet sich aus den Gemeinsamkeiten mit physischen Drahtmodellen ab: Sie bestimmen die Umrisse eines Objektes und gewähren von jeder Seite die Durchsicht. Ist dies für einfache Modelle oft ein Vorteil, so kann es sich für komplexere Gebilde schnell als Nachteil erweisen, da die Lesbarkeit nicht mehr gegeben ist.

Flächenmodelle stellen Objekte als ein geordnetes Set von Flächen im dreidimensionalen Raum dar. Flächenmodellierer verwendet man zur Generierung und Darstellung dreidimensionaler Objekte. Sie erlauben es, verdeckte Linien in Projektionen zu berechnen und auf dem Bildschirm zu unterdrücken. Dabei werden jeweils mindestens drei oder mehr Eckpunkte zu einer Fläche zusammengefaßt. Nur bei Flächen mit drei Eckpunkten ist garantiert, daß diese in einer Ebene liegen, bei mehr als drei Eckpunkten können windschiefe Flächen entstehen. Für die Simulation von Material und Licht auf Flächen ist es für die mei-

sten Algorithmen wichtig, in welcher Reihenfolge die Eckpunkte eingegeben wurden, denn dadurch wird die Vorder- und Rückseite einer Fläche definiert. Werden zwei Flächen miteinander verschnitten, so erscheint bei einfachen Flächenmodellierern die Verschneidungslinie oft nicht, wodurch optische Täuschungen entstehen können. Aus diesem Grund ist es auch schwierig, mit Flächenmodellierern Schnitte durch Gebäude zu legen [Harrington 1983].

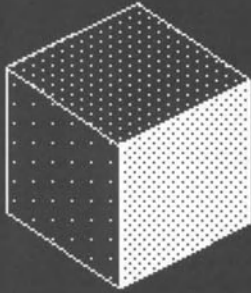
Volumenmodelle stellen dreidimensionale Objekte durch Volumen dar. Die Datenstruktur der Volumenmodellierer erlaubt die Anwendung der Booleschen Operationen Vereinigung, Subtraktion und Verschneidungsmengenbildung sowie die Berechnung von Volumen, Schwerpunkt und Oberflächen [Mäntylä 1988]. Solid Modeling ist die anspruchsvollste und auch rechenintensivste Modellierart. Volumenmodellierer eignen sich als intuitives Werkzeug zur Generierung, Manipulation und Darstellung volumetrischer Modelle. Solid Modeling behandelt Objekte als wirkliche Körper mit beliebig komplexer Oberfläche, die auch Öffnungen und Durchdringungen enthalten können. Schneidet man ein Solid, so entsteht wiederum ein geschlossenes Solid. Es ist natürlich, mit Volumen zu arbeiten und mit physischen Modellen zu entwerfen. Ein Volumenmodellierer baut auf den Erfahrungen des physischen Modells auf und verläßt sich auf die Grundlagen, die Designer - meist in der Industrie - damit gesammelt haben. Der Hauptunterschied zwischen dem physischen Modell und Solid Modeling ist, daß durch das Computerprogramm die Struktur des Entwurfs mit der Generierung formaler Repräsentationen entstehen kann. Der Volumenmodellierer behält stets die Hierarchie präsent, in der die Elemente (Primitives) und die Operationen (Operations) festgelegt sind.

Element-Typ

Beispiel

Anwendung

Refvem und van der Mark, 1992



Pixel - Point



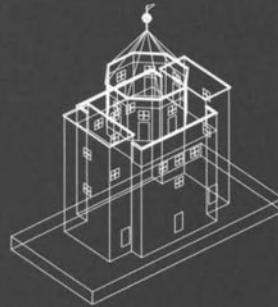
Gescannte Handzeichnung



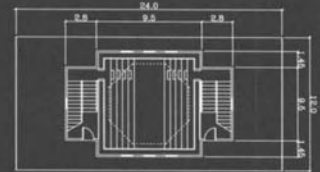
Image Processing



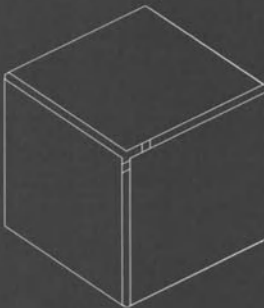
Vektor -Line



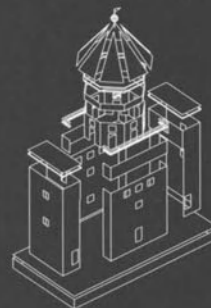
Dreidimensionales Drahtmodell



Zweidimensionales Flächenmodell



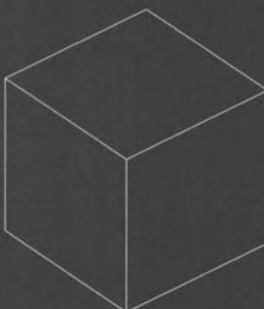
Fläche - Face



Explodiertes Flächenmodell



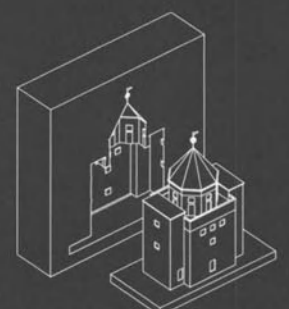
Dreidimensionales Flächenmodell



Objekt - Volume



Explodiertes Volumenmodell



Subtraktion mit Volumenmodellen

Instrument: Programmieren I - Traditionelle Programme

Anwendung: CAD Programme, Datenbanken

Verwandte Gebiete: Formalisierte Sprache

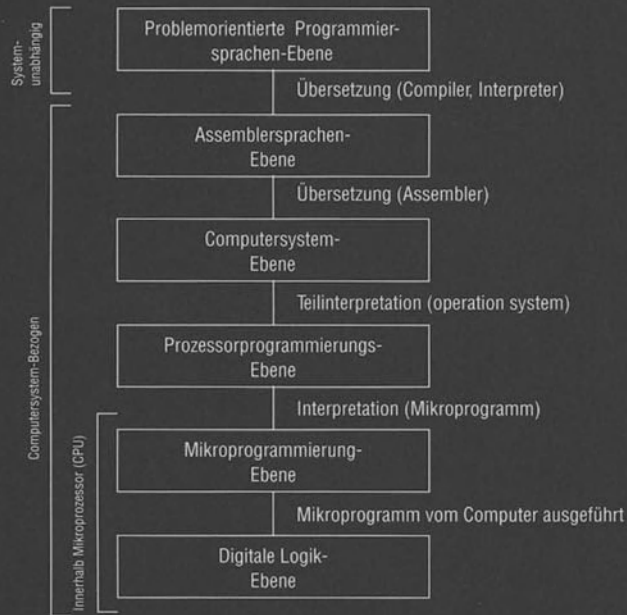
Jedes Programm besteht aus einem Text, der vom Computer interpretiert und ausgeführt wird. Der Text beinhaltet Berechnungen oder steuert Prozesse. Er ist in einer Programmiersprache geschrieben, der eine formale Definition zugrunde liegt. Das bedeutet, daß die so definierte Sprache bestimmten Regeln für Textzusammensetzung und Abläufe (Syntax und Semantik) folgen muß. Mit einer formalisierten Sprache kann man Abläufe und Strukturen am besten beschreiben. Eine Vielzahl verschiedener Programmiersprachen weist eine große Variation in Gestalt und Fähigkeit auf. Die Programmiersprachen lassen sich in diverse Ebenen aufteilen: Man spricht von der Hierarchie der Programmiersprachen. Die Einordnung dieser Sprachen hängt davon ab, wie nahe sie der Sprache (Code) ist, welche die Recheneinheit und die Peripherie direkt verstehen.

Auf den unteren Ebenen sind Maschinensprache und Assemblersprache angesiedelt. Sie umfassen die einfachen und elementaren Befehle zur Programmierung von Computern. Jedes Programm in einer höheren Programmiersprache wird schließlich immer in diese untere Ebene übersetzt, entweder durch Compiler oder Interpreter. Höhere Programmiersprachen sind einfacher zu verwenden, da sie durch Zusammensetzung einfacher Befehle komplexe Manipulationen erlauben. Ein Aspekt vieler höherer Programmiersprachen wie Pascal oder C ist der prozedurale Ansatz. Dieser setzt voraus, das man ein Problem in verschiedene Teilprobleme gliedern kann. Teilprobleme können wiederum solange unterteilt werden, bis man das Gesamtproblem mittels bestehender Befehle lösen kann. Auch für ähnliche Teilprobleme können gemeinsame Lösungen gefunden werden. Man spricht dabei von Funktionen und Prozeduren als Bestandteilen des Lösungsablaufs. Damit lassen sich klar strukturierte Probleme sowie sol-

che, bei denen numerische Berechnungen im Vordergrund stehen, gut angehen.

Ein anderer Ansatz für viele Programmiersprachen ist die symbolische Programmierung. Dabei steht nicht der Ablauf des Programms im Vordergrund, sondern das abstrakte Konzept des Problems und die Zusammenhänge der Daten. Ziel ist die Manipulation dieser Zusammenhänge. Vielfach verschmelzen dabei Daten und Programm zu einer Einheit. Typische Beispiele dieser Richtung sind Lisp, verschiedene Programme für regelbasierte Systeme, sowie Prolog. Aufbauend auf diesen konventionellen Programmiersprachen wurden und werden ständig neue Sprachen entwickelt, die für bestimmte Arten von Problemlösungen geeignet sind. Die objektorientierte Programmierung zum Beispiel vereinigt Eigenschaften prozeduraler und symbolischer Programmierung.

Traditionelle Programmiersprachen sind die Basis der meisten heute benutzten Software. Sie ermöglichten erst die Entwicklung verschiedener Anwenderprogramme. Auch im Bereich der Architektur liegen dem überwiegenden Teil der CAD- und Berechnungsprogramme noch immer konventionelle Sprachen zugrunde. Je größer die Zahl der Anwendungen und je umfangreicher die Programme, desto schwieriger sind tiefgreifende Änderungen in den Applikationen zu verwirklichen. Dies ist ein Problem einiger CAD Programme, die im Verlauf vieler Jahre entstanden. Grundsätzlich sind verschiedene Computersprachen miteinander kompatibel. So bestehen heute große CAD-Programme oft aus einer Kombination von Fortran und C. In der Vergangenheit lernten Architekturstudierende als erstes eine Sprache wie Fortran oder Pascal. Die Lehrenden unternahmen Versuche, Zusammenhänge zwischen Programmen und Graphik zu verdeutlichen [Mitchell 1987].



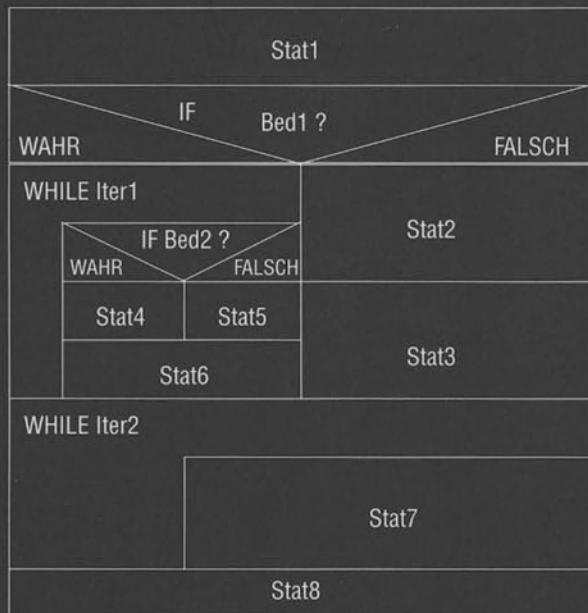
Hierarchie der Ebenen in Programmiersprachen

```

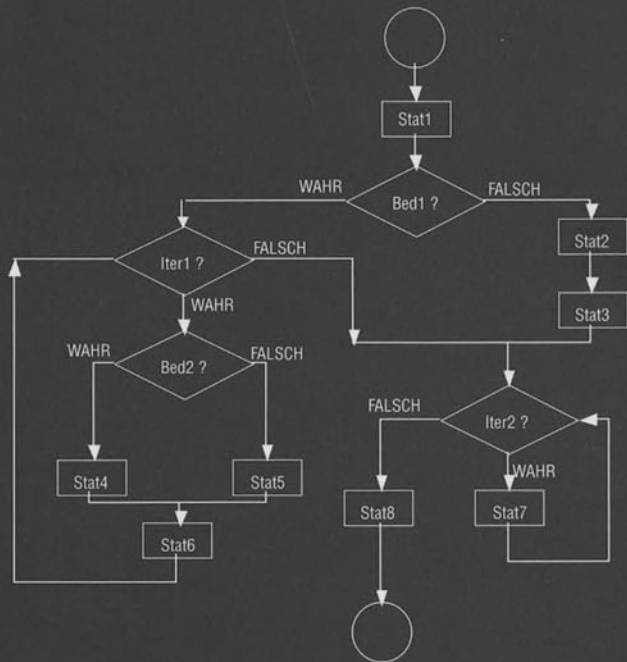
BEGIN
  Stat1,
  IF NOT Bed1
    THEN
      BEGIN
        Stat2; Stat3;
      END;
    ELSE
      WHILE Iter1 DO
        BEGIN
          If Bed2
            THEN Stat4;
            ELSE Stat5;
          Stat6,
        END;
      WHILE Iter2 DO Stat7;
      Stat8;
    END;
  END;

```

Teil eines Pascal-Programms



Strukturierter Programmablauf analog zum obigen Code



Das Flußdiagramm zeigt den Ablauf des Programms. Kurmann, 1993

Instrument: Programmieren II - Wissensbasierte Programme

Anwendung: Design-Automatisierung, Bauvorschriften

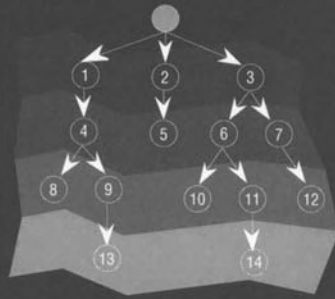
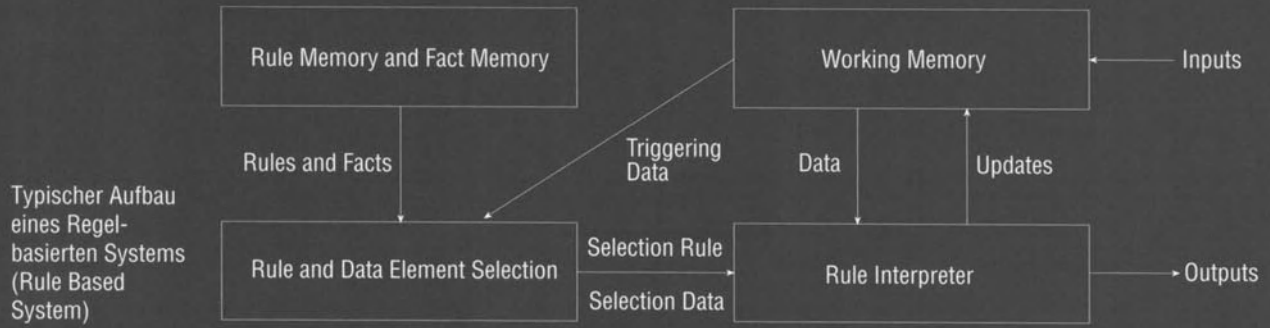
Verwandte Gebiete: Formengrammatiken

Wissensbasiertes Programmieren unterstützt die explizite Repräsentation menschlichen Wissens auf einem bestimmten und abgegrenzten Gebiet. Es hat die Entwicklung von wissensbasierten Systemen (Knowledge-Based Systems, KBS) zum Ziel. In traditionellen Programmen werden das Wissen über ein bestimmtes Gebiet und die Anweisungen, mit diesem Wissen umzugehen, in einem einzigen Programm zusammengefaßt oder vermischt. Dies erschwert die Untersuchung und die Veränderung des Wissens und des Verarbeitungsprozesses (Reasoning Process) und kann zudem unerwünschte Nebeneffekte zwischen den beiden erzeugen. Wissensbasiertes Programmieren dagegen behält die Trennung zwischen Wissen und seiner Anwendung, also dem spezifischen Verarbeitungsprozeß, bei. Dies ermöglicht die Entwicklung von wissensbasierten Programmen, die nicht nur für eine bestimmte Problemlösung einsetzbar sind. In diesem Aspekt weisen wissensbasierte Programme gewisse Ähnlichkeiten mit menschlichen Experten auf, die ihr Wissen und ihre Erfahrung einsetzen, um auch in unerwarteten Situationen richtig reagieren und handeln zu können.

Bei der Entwicklung von wissensbasierten Systemen stehen zwei Gesichtspunkte im Mittelpunkt: die Repräsentation des Wissens und der Verarbeitungsprozeß. Im Verlauf der Zeit haben sich verschiedene Wissensrepräsentationen entwickelt, die jedoch alle die Übersetzung von Wissen über Objekte, deren Eigenschaften und Beziehungen untereinander in Computerprogramme zum Inhalt haben. Einige der bekannteren Wissensrepräsentationen sind die Verkapselung von Wissen in Klassen (Classes) und Variationen (Instances), Rahmen (Frames) und Schemata (Schema), sowie Regeln (Rules) und Produktionen (Productions). Verarbeitungsprozesse wirken sich auch

auf das Wissen aus, das in einem bestimmten Umfeld (Context) zur Verfügung steht. Typischerweise arbeiten wissensbasierte Systeme in folgenden Schritten: Am Beginn besteht eine Ausgangslage an Information (Initial State of Information). Von dieser aus geschieht die Transformation der Information von einem Zustand (State) in einen weiteren. Der Prozeß bewegt sich schließlich auf eine Zielvorstellung (Goal State) als erwünschte Lösung eines Problems hin. Verschiedene Verarbeitungsprozesse kommen dabei zum Einsatz: Vorwärtsdeduktion (Forward Chaining), Rückwärtsdeduktion (Backward Chaining), Tiefensuche (Depth First), Breitensuche (Breadth First), Maximumsuche (Hill Climbing) und andere [Akin 1986, Lenart 1991].

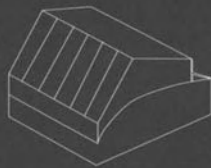
Die Anwendungen von wissensbasierten Programmen in der Architektur sind inzwischen zahlreich [Coyne 1990]. Sie reichen von einfachen Produktionssystemen zur Auswahl und Dimensionierung von Lüftungsanlagen bis zum Versuch, ganze Entwurfsprozesse mit wissensbasierten Programmen zu simulieren [Rychener 1988]. Eine besondere Art der wissensbasierten Systeme sind die Expertensysteme (Expert Systems), auf die in der Mitte der achtziger Jahre große Hoffnungen für die Lösung praktischer Probleme gesetzt wurden. Sie lassen sich unter Verwendung von Expertensystem-Schalen (Expert System Shells) programmieren, die kein Wissen, dafür aber die verschiedenen Verarbeitungsprozesse beinhalten. Nach der Formulierung von Expertenwissen lassen sich dann Klassifizierungs- oder in begrenztem Maß auch Entwurfsaufgaben durchführen [Schmitt 1989b, Schmitt 1988b]. Die großen Hoffnungen auf die Leistungen der Expertensysteme wurden noch nicht erfüllt. Doch haben in diesen Systemen verwendete Techniken in viele Programme Eingang gefunden.



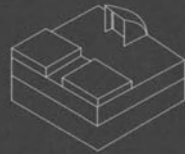
Breadth-first Processing



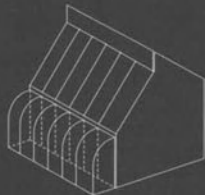
Depth-first Processing



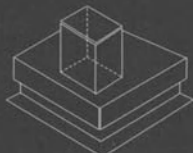
IF site is in Cool Climate
 THEN design partly underground massing
 AND enable snow accumulation to stand against walls or on roofs
 AND enable internal heat zoning with air-locks at entry ways



IF site is in Hot Dry Climate
 THEN design for natural cooling and heating
 AND provide interior courtyards
 AND partly underground building



IF site is in Temperate Climate
 THEN design for exposure to sun and wind
 AND provide greenhouses
 AND provide passive solar-heat devices



IF site is in Hot Wet Climate
 THEN design for maximum exposure to breeze
 AND provide interior ventilation courtyards

Beispiele für Entwurfsregeln, die sich aus klimatischen Verhältnissen ableiten. Dave, 1991

Instrument: Programmieren III - Objektorientierte Programme

Anwendung: Hierarchische Architekturmodelle

Verwandte Gebiete: Sprache, graphische Objekte

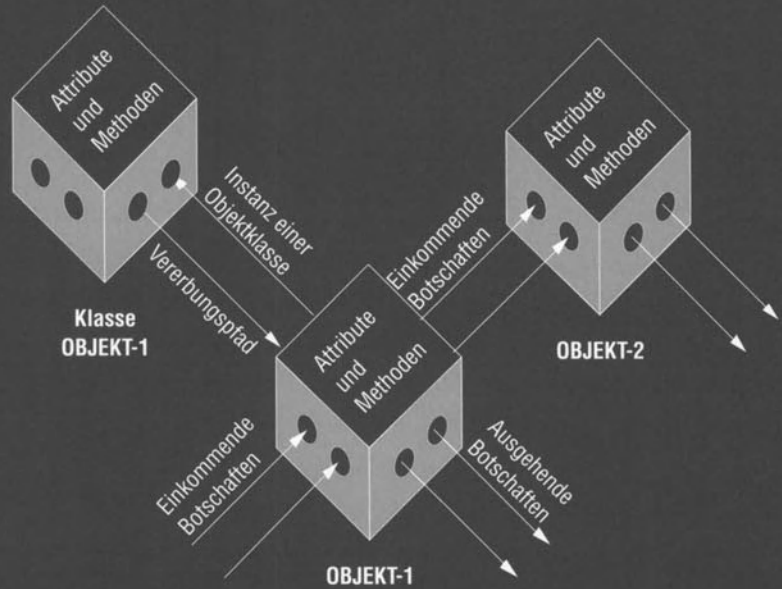
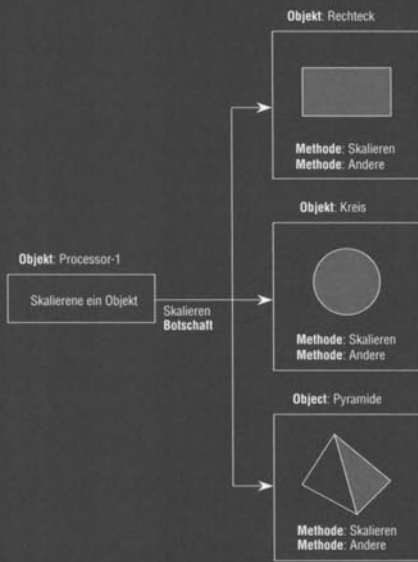
Die objektorientierte Programmierung (OOP) ist ein relativ neuer Ansatz in der Entwicklung von Software [Cox 1987]. Das Ziel ist, eine bessere Art der Repräsentation von Sachverhalten zu erreichen und die Entwicklung von Programmen zu optimieren. OOP konzentriert sich auf die Organisation von Programmablauf und Daten. Das Objekt bildet die Einheit, in der sich Daten und funktionaler Ablauf des Programms vereinen. Die Attribute (Daten) eines Objekts sind die internen Variablen und Zustände, die das Objekt einnimmt. Methoden sind die Operatoren und Funktionen, die ein Objekt ausführen kann. Eine Funktion ist an ein Objekt gebunden.

Jedes Objekt ist eine Variation einer Objektklasse, die das allgemeine Verhalten und die Daten eines Typs von Objekten beschreibt. Im Gegensatz zum Konzept der Datentypen in konventionellen Programmiersprachen erweitert OOP den Begriff des Typs auch auf die Methoden. Hierarchie und Vererbung ermöglichen einen einfachen und logischen Aufbau der Objektklassen, wobei das Ausnutzen von Gemeinsamkeiten und somit die Reduktion des Entwicklungsaufwandes für das Programm zentral sind. Die Objektklassen werden gemäß ihrer Eigenschaften in hierarchischen Strukturen implementiert. Jede Klasse erbt das Verhalten und die Daten ihrer Elternklasse und kann um eine verfeinerte Funktionalität erweitert werden. Die Einkapselung von Informationen (Information Hiding) garantiert die Eigenständigkeit und die Unabhängigkeit von Objekten. Dieser sehr wichtige Aspekt moderner Programmiersprachen garantiert eine klare Schnittstelle zwischen Programmteilen. Der Zustand (die internen Daten) eines Objekt können so beschrieben werden, daß sie von außen nicht sichtbar sind. Um einem Objekt zu erlauben, den Zustand eines anderen Objektes abzufragen, muß eine Funktion auf

dem Objekt definiert werden. Diese Abfrage geschieht über Botschaften (Messages), die zwischen den Objekten als Kommunikations-Schnittstelle dienen. Vielfach sind diese Konzepte auf bestehenden, konventionellen Programmiersprachen als Bibliotheken (Libraries) implementiert, wie beispielsweise C++ auf C oder Objectpascal auf Pascal. In Benutzerschnittstellen und graphischen Applikationen wurde OOP schon früh kommerziell eingesetzt.

Das objektorientierte Programmierparadigma ist ein grundlegendes Werkzeug für restriktions-basiertes Modellieren (Constraint Based Modeling). Doch bereits vor seiner Erfindung führte Sketchpad die Verwendung von Restriktionsmanagement (Constraint Management) in interaktive graphische Systeme ein [Sutherland 1963]. Die Möglichkeiten des interaktiven Constraint-Based-Modeling verbreiteten sich erst mit Bornings ThingLab-Arbeit in den frühen achtziger Jahren [Borning 1981]. ThingLab erweiterte die objektorientierte Sprache Smalltalk, damit sie die interaktive Definition und den Unterhalt von Constraints zwischen Objekten erlaubt.

OOP ist speziell für die Modellierung von Hierarchien und vererbbaaren Strukturen einsetzbar, was für viele Anwendungen in der Architektur zutrifft. So lassen sich die meisten der traditionellen hierarchischen Architekturbeschreibungen direkt als OOP Modell umsetzen. Dies funktioniert solange, wie die syntaktischen Beziehungen der Teile zueinander bestehen bleiben. Beliebte Beispiele sind die Modellierung von Möbeln, die aus auswechselbaren Einzelteilen bestehen, oder von Bautelementen, die sich in Einzelteile zerlegen lassen. OOP ist weniger nützlich in der Modellbildung, wenn topologische Änderungen vorgenommen werden müssen, welche dynamisch die Grundstruktur des Modells verändern.



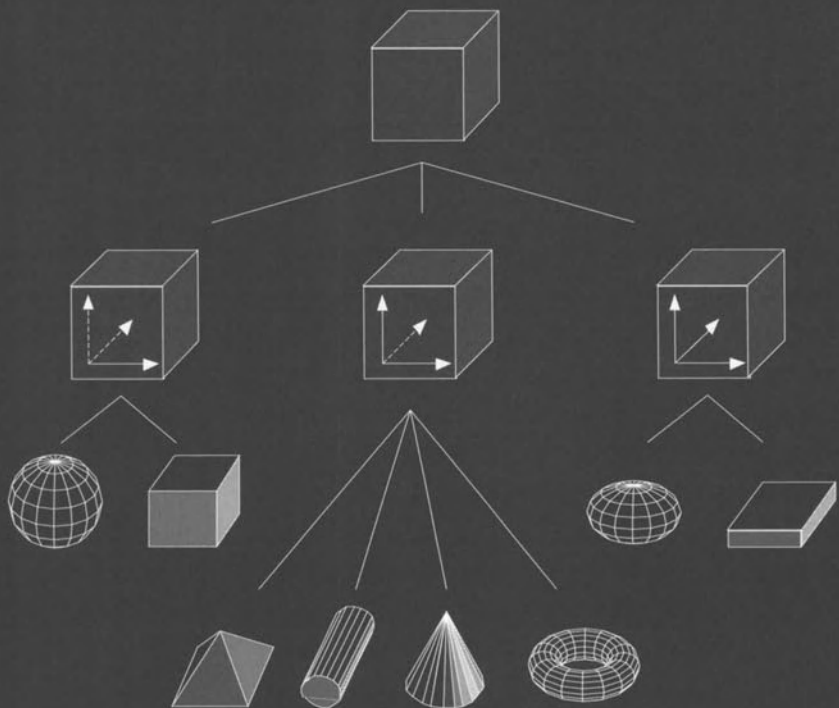
Objektverknüpfungen

Objekte und Datenaustausch zwischen Objekten

Stufe 1:
 Allgemeine 3D-Objektklasse
 Attribute: Position, Farbe
 Methoden: Verschieben, Farbe abfragen und ändern

Stufe 2:
 Parametrisierte 3D-Objektklassen
 Attribute: Parameter (Dimensionen)
 Methoden: Parameter abfragen und verändern

Stufe 3:
 Spezifische 3D-Objektklassen
 Methoden: Volumen berechnen, Zeichnen.
 Kurmann, 1993



Instrument: Generieren I - Formengrammatiken

Anwendungen: Analyse und Synthese von Design Syntax

Verwandte Gebiete: Sprache

Die Anwendung von Formengrammatiken in der Architektur basiert auf der Annahme einer Analogie zwischen gesprochener Sprache und Architektur. Bietet die Sprache grammatische Regeln, nach denen Worte Sätze bilden, so existieren auch grammatische Formregeln, nach denen ein Vokabular von Entwurfs-elementen zu einem Entwurf kombiniert werden kann. Formengrammatiken sind das graphische Äquivalent der Produktionssysteme und werden auch als solche implementiert. Eine graphische Regel entspricht einer geometrischen Transformation, die eine Wenn- oder Bedingungsseite (Left Hand Side, LHS) und eine Dann- oder Aktionsseite (Right Hand Side, RHS) hat. Sowohl LHS als auch RHS sind geometrische Formen. Bei Anwendung der graphischen Regel wird die LHS durch die RHS ersetzt.

Die Anwendung von Formengrammatiken in der Architektur geht von der Möglichkeit aus, Architektur als Sprache zu sehen. Eine Sprache besteht aus "einer Reihe von Regeln, nach denen Symbole zur Darstellung komplexerer Bedeutungen verknüpft werden" [McKim 1980, dt. v. Verf.]. Die Hauptbestandteile einer Sprache sind Vokabular, Syntax, Semantik, Kontext und Stil [Winograd 1983]. Auch Architektur besitzt diese Merkmale. Der Nachweis von Vokabular und Syntax an bestehenden Gebäuden ist spätestens durch die Arbeiten von Stiny und Mitchell über Palladio [Stiny 1978, Stiny 1980], nach den Untersuchungen von Koning und Eizenberg über Frank Lloyd Wright [Koning 1981] und durch die Arbeiten von Flemming über Queen Anne-Häuser [Flemming 1987] erbracht.

Grammatiken sind hervorragende Instrumente zur Lösung lokaler Probleme syntaktischer Natur, bei denen räumliche und formale Aspekte im Vordergrund stehen. Die Probleme mit dem Einsatz von Grammatiken in der Architektur beginnen, wenn man die verbleibenden Bereiche betrachtet: Wenig Hilfe bieten die Grammatiken

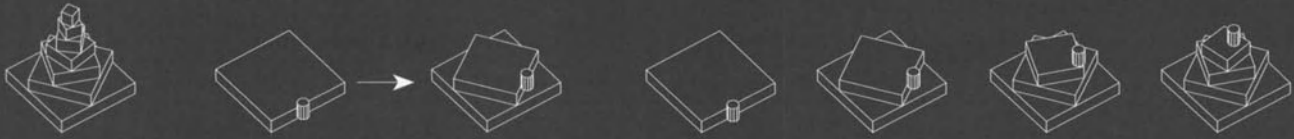
in der Definition von Semantik, Kontext und Stil, es sei denn, diese ließen sich jeweils auf syntaktische Definitionen zurückführen.

Ein praktisches Beispiel für den Einsatz von Formengrammatiken ist das Programm BAU [Kramel 1992], entstanden in Zusammenarbeit zwischen Architekt und Wissensingenieur. Hintergrund ist die Untersuchung architektonischer Module im zunächst zweidimensionalen funktionalen Raum und danach im dreidimensionalen und geometrischen Raum. Die insgesamt fünf-Gebäudefunktionen lassen sich auf einem 3x3-Raster vielfältig anordnen, der dreidimensionale Lösungsraum umfaßt gar über 500 Millionen möglicher Kombinationen. Um diese für Grammatiken typischen kombinatorischen Explosionen zu vermeiden, werden begrenzte Regeln und Gewichtungen eingeführt.

Ein weiteres Beispiel ist die Idee der Verwendung von Froebelschen Bauklötzen in der Architektur. Dabei werden die Markierungen nicht mehr auf Formen in der Ebene gesetzt, sondern auf Körper im Raum. Zusätzlich ergibt sich die Möglichkeit, die Regeln graphisch zu definieren und danach als benannte Regel zur späteren Verwendung abzuspeichern. Aus den so definierten Regeln lassen sich dann ganze Sätze formen, die wiederum Teil der durch die Grammatik definierten Sprache sind [Schmitt 1991].

Wenige Forscher haben bisher versucht, Formengrammatiken für moderne Bauten zu definieren. Eine Ausnahme ist Flemming, der eine Grammatik für die Casa Giuliani Frigerio entwickelte [Flemming 1981]. Spätestens zu dem Zeitpunkt, an dem man selbst eine Formengrammatik definiert hat und beginnt, mit ihr ein neues Gebäude zu entwerfen, tritt die Frage nach dem Sinn eines solchen Vorgehens auf. Denn zu diesem Zeitpunkt wird klar, welche wichtigen Aspekte des Entwurfs eine Grammatik noch nicht behandeln kann.

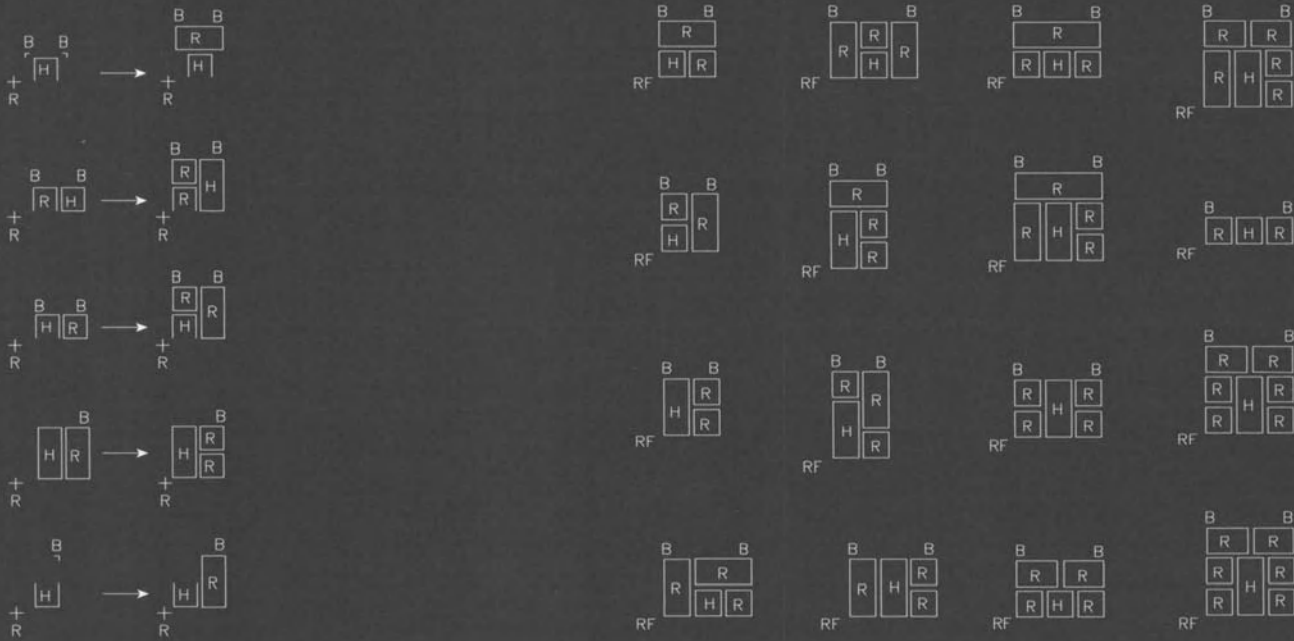
Formengrammatiken für die Erzeugung einfacher geometrischer Objekte



Komposition

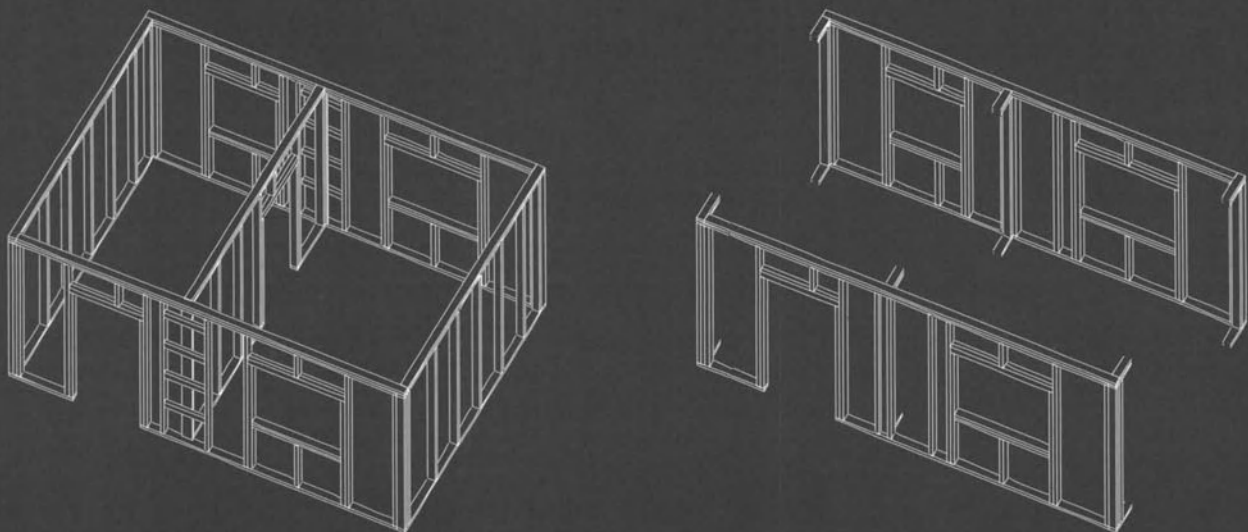
Formenregel

Anwendung der Regel



Regeln für eine Formengrammatik zur Platzierung des Wohnraums in einer viktorianischen Villa. Fleming 1988

Schematische Grundrisse, die mit Formengrammatiken generiert wurden. Fleming, 1988



Formengrammatiken in der Konstruktion: Paneele in Holzrahmenbauweise, die mit einer Grammatik generiert wurden. Shih, 1993

Instrument: Generieren II - Fraktale

Anwendungen: Naturforschung, Bildkompression, Szenensimulation

Verwandte Gebiete: Mathematische Beschreibung der Natur.

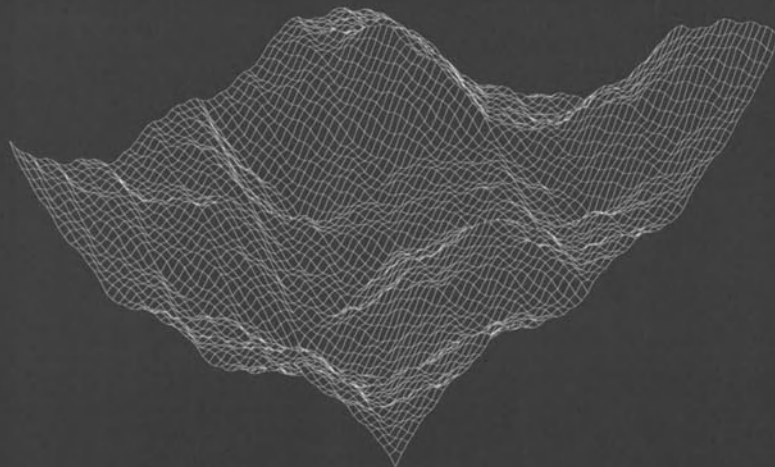
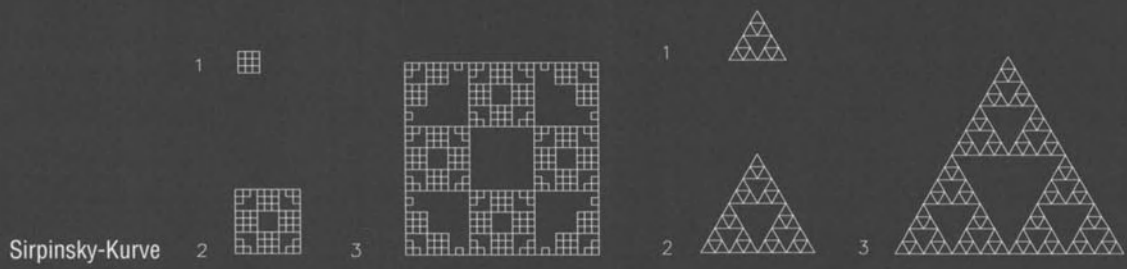
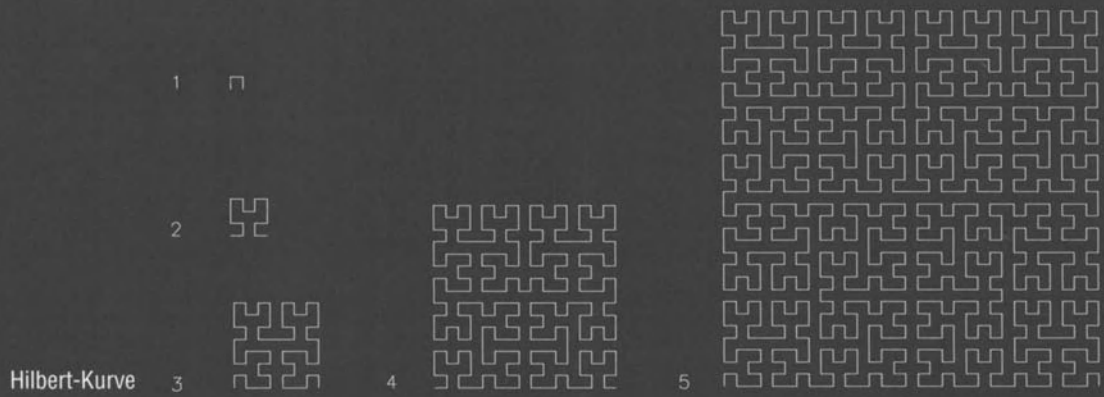
Die fraktale Geometrie erlaubt es, natürliche Formen mathematisch zu beschreiben und chaotische Systeme bildlich zu veranschaulichen. Ein bekanntes Fraktal ist jede Küstenlinie. Betrachtet man einen immer kleineren Ausschnitt der Küste, zeigen sich mit jedem Schritt mehr Buchten und Halbinseln. Mit jedem Fels oder Sandkorn wiederholen sich dieselben, selbstähnlichen Muster, und die Gesamtlänge wächst gegen Unendlich. Neben der Selbstähnlichkeit haben Fraktale die Eigenschaft der Nichtlinearität: Eine Küstenlinie kann nicht mit den Elementen der euklidischen Geometrie, der eindimensionalen Geraden oder der zweidimensionalen Fläche beschrieben werden. Die Dimension der Küstenlinie ist größer als eins, aber kleiner als zwei. Sie entspricht damit einer gebrochener Zahl zwischen eins und zwei, daher der Name Fraktal. Fraktale sind durch die fraktale Dimension vergleichbar. So unterscheiden sich etwa Bäume durch ihre Dimension zwischen zwei und drei. Man kann Fraktale als Untermenge (Subset) der Formengrammatiken betrachten, denn es genügt eine rekursiv angewendete Formenregel, um komplexe Gebilde zu generieren [Peitgen 1986, Peitgen 1988].

Ein bekannter Ansatz ist Mandelbrots Arbeit über die fraktale Geometrie der Natur, in der die Relevanz rekursiv definierter Formen und Kurven für viele Wissenschaftsbereiche deutlich wird [Mandelbrot 1982]. Wie Mandelbrot mit der Beschreibung von Wolken, Bäumen, Küstenlinien und anderen Formen nachweist, sind fast alle natürlichen Formen durch fraktale Algorithmen darstellbar. Er findet die Architektur der Beaux Arts-Periode reich an fraktalen Aspekten, nicht aber die Bauten Mies van der Rohes. Diese auf den ersten Blick einleuchtende Beobachtung kann jedoch nicht unkritisch übernommen werden, lassen sich doch gerade auch für Mies van der Rohes frühe Entwürfe - wie im Fall des Backstein-Landhauses (1923), des Liebknecht-

Luxemburg-Denkmal (1926) und sogar des Barcelona-Pavillons (8/1929) fraktale Entwurfaspekte nachweisen. Gleiches gilt für Projekte von Christopher Alexander, Daniel Libeskind und Simon Ungers, um nur einige zu nennen. Die Untersuchung der fraktalen Geometrie der Architektur ist spätestens seit den Bauten von Sullivan und Eisenman ein interessantes, noch offenes Forschungsthema. Peter Eisenman zum Beispiel behauptet von sich selbst, ein 'fraktaler Denker' zu sein. In verschiedenen seiner Bauten konnten wir in der Tat fraktale formale Aspekte nachweisen [Schmitt 1988a, S. 141].

Rekursives Suchen scheint besonders im Frühwerk von Architekten stattzufinden, während das spätere, reifere Werk sich auf Teilaspekte dieser Erfahrungen konzentriert, sie verallgemeinert und oft ins Monumentale steigert, wobei eine fortschreitend Loslösung von der 'Natur' zu beobachten ist. Einen Schritt weitergehend, könnte man behaupten, daß das Fraktale, Rekursive sich als lokales Gleichgewicht herausbildet, während das 'euklidisch' Geordnete einer reduzierten und daher verständlicheren globalen Ordnung entspricht. Umgekehrt läßt sich vermuten, daß lokales und naturnahes Bauen und Entwerfen sich fraktalen Denkweisen annähern wird, während das Entwerfen im 'euklidischen' Raum immer nur eine begrenzte Formensprache benutzt und daher im gewollten Gegensatz zur 'Natur' steht. Mit der völligen Wandlung des Naturbegriffs ist somit eine stärkere Betonung fraktaler - und daher im neuen Sinn naturverwandter Formen - in der Architektur und Stadtplanung denkbar.

Die enge Beziehung zwischen Fraktalen und Formengrammatiken gibt Anlaß zur Untersuchung in Entwurfsklassen. So fanden an der Ohio State University und an der UCLA wiederholt Studios mit diesem Inhalt statt [Knight 1992].



Fraktale Landschaft, Shih 1990

Instrument: Parametrisierung

Anwendungen: Computer Aided Design, Produktion von Bauteilen, Geschichtsforschung

Verwandte Gebiete: Geometrie, Ähnlichkeit

Parameter sind Größen, mit denen Objekteigenschaften innerhalb gesetzter Grenzen variiert werden können, ohne daß sich der Charakter der Objekte dabei grundlegend ändert. Die Topologie des Objektes bleibt erhalten, seine Geometrie kann sich insgesamt oder in Teilbereichen ändern. Ist ein Objekt einmal parametrisiert, lassen sich unendlich viele parametrische Variationen daraus ableiten. Dies bedingt Effizienz, denn die einzelnen Objekte sind nicht separat zu speichern, sondern lediglich die Parameter, nach denen sie sich unterscheiden. Das ursprüngliche Objekt, die Parameter, ihre Begrenzungen (Ranges) und die notwendigen Herstellungsmethoden bilden so eine Beschreibung für eine unendliche Vielfalt ähnlicher Objekte. Besitzt ein Objekt mehrere Parameter, so können Interferenzen auftreten, die entweder manuell oder automatisch zu entschärfen sind. Andererseits kann die absichtliche oder zufällige überproportionale Betonung bestimmter Parameter zu neuen Sichtweisen des Objektes führen. Parametrisierung wird entweder bei der Definition des Objektes angegeben oder automatisch durch Methoden wie fallbasiertes Schließen erreicht.

Attribute, die sich an Objekten als Parameter eignen, sind unter anderem Farbe, Material oder Größe. Parameter, welche die Größe bestimmen, unterliegen oft speziellen Begrenzungen, denn ohne diese Begrenzungen könnten sich die Qualität und der Gesamtcharakter des Objektes bei unkontrollierter Veränderung der Parameter so stark wandeln, daß das Ergebnis mit dem ursprünglichen Objekt kaum mehr etwas gemein hat. Parametrisierung ist ein wichtiges Instrument, mit dem Prototypen erfolgreich an neue Anforderungen angepaßt werden können. Parameter sind besonders hilfreich bei der Definition und dem Unterhalt standardisierter Bauteile, wie Fenster, Türen und Treppen. Die meisten CAD-Programme unterstützen die Parametri-

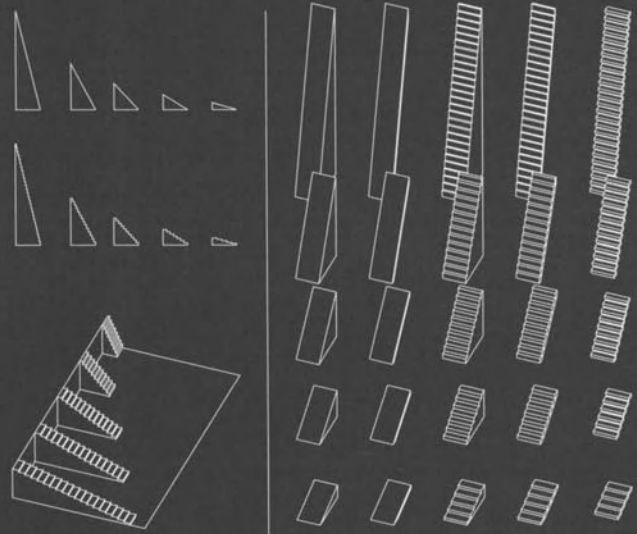
sierung von Objekten bereits als Grundfunktion. Ausgestattet mit einer intuitiven Benutzeroberfläche, wird sie sich zu einem der wichtigsten CAD-Instrumente entwickeln.

Die Wahl der Parameter ist ein wichtiger Anfangsschritt. Bei Variationen eines menschlichen Gesichtes zum Beispiel wird die Lage der relevanten Teile, wie Nase, Kinn, Augen, Mund und Ohren, in einem Raster festgelegt. Durch Verschiebung dieses Rasters kommen die verschiedensten Gesichtsausdrücke zustande. Bei der Animation dreidimensionaler Gesichtsmodelle sind die wichtigsten Punkte ebenfalls parametrisiert und verformen das sie umgebende dreidimensionale Netz im Fall ihrer Veränderung. Bei einer genügenden Zahl von Netzpunkten läßt sich so jede Form in eine beliebige andere Form überführen. In der Architektur sind Grenzfälle von Interesse, an denen die Parametrisierung allein nicht weiterhilft. So lassen sich parametrisierte Objekte mit Regeln ausstatten, die den Benutzer vor der Verletzung von Baugesetzen oder statischen Fehlern schützen. In ein parametrisiertes Treppenobjekt lassen sich Regeln über die maximale und minimale Zahl der Auftritte, das erlaubte und erwünschte Steigungsverhältnis, die erforderliche Breite und das Einfügen von Podesten einbauen. Das Treppenprogramm macht die Anwender lediglich auf Verletzungen der Regeln aufmerksam und leistet ansonsten alle Konstruktionsarbeit.

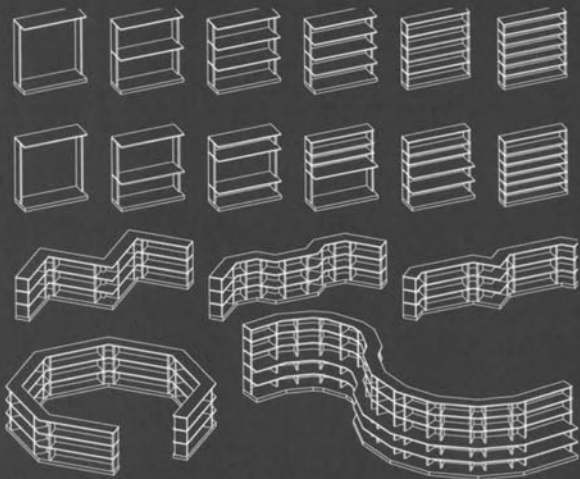
Ein parametrisiertes Programm zur Herstellung von Büchergestellen beispielsweise kann weitere Aufgaben wahrnehmen. Wenn die Anwender graphisch oder numerisch die wichtigsten Attribute definiert haben, beginnt das Programm mit der Konstruktion des dreidimensionalen Objektes und macht die entsprechenden Datenbankberechnungen über die unterzubringenden Bücher.



Parametrisierte Gesichter, Shih, 1992



Parametrisierte Büchergestelle, Refvem, 1992



Parametrisierte Treppen, Engeli, 1992



Parametrisierte Häuser, Refvem, 1993

Instrument: Objektorientiertes Modellieren I - Typen und Variationen

Anwendung: Entwurf

Verwandte Gebiete: Objektorientiertes Programmieren

Als Typ bezeichnen wir ein Objekt, das sich durch Form, Funktion oder beides eindeutig identifizieren läßt. Wir beziehen uns damit nicht primär auf die Typendiskussion in der Architektur, sondern definieren einen Typ in Anlehnung an William Mitchells Diskussion dieses Themas, besonders auf seine Beschreibung der essentiellen und zufälligen Eigenschaften eines Typs [Mitchell 1990a, S. 86]. In der Architektur oft wiederkehrende Elemente können damit durch Variation eines Typs abgeleitet werden. Obwohl die Form oder der Inhalt in den meisten Fällen einen Typ definieren, beziehen wir auch andere Attribute, wie Farbe, Transparenz oder Funktion, ein. In einem bestimmten Einsatzgebiet kann die Farbe das Essentielle sein, das einen Typ definiert, in einem anderen Bereich kann es der Wärmedurchgang sein, der den Typ bestimmt.

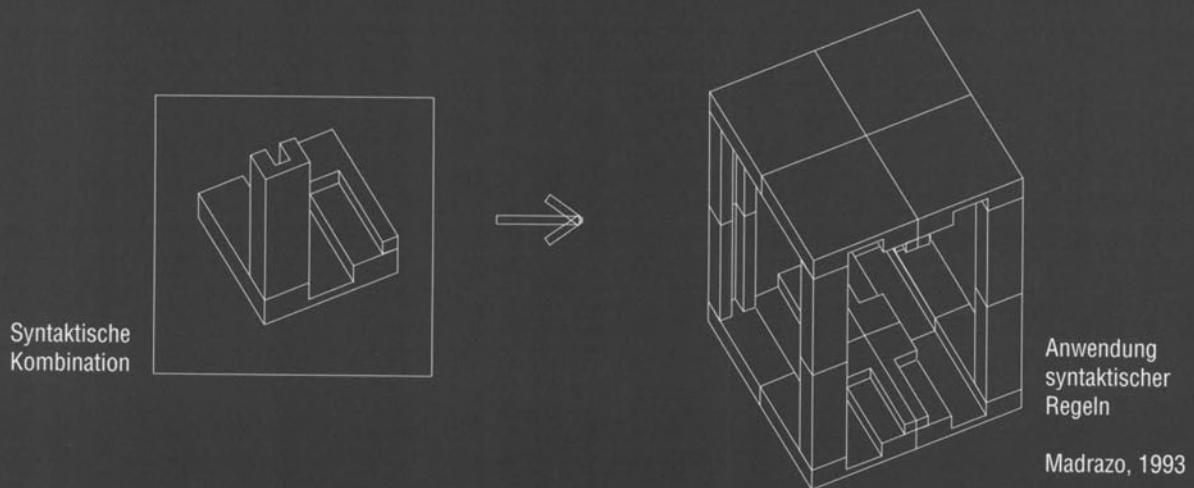
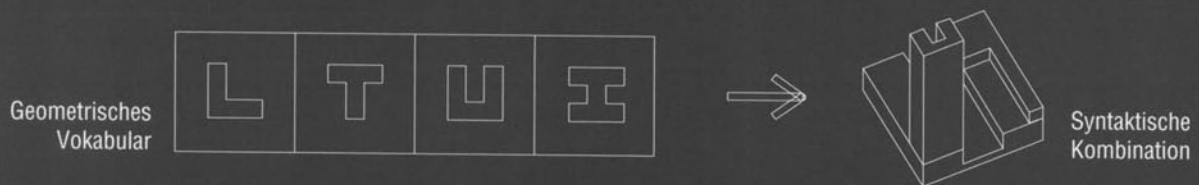
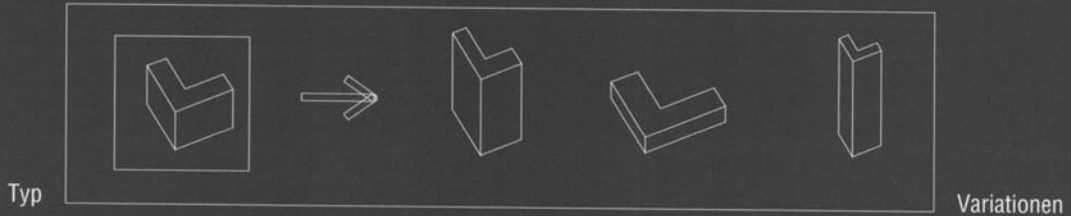
Kommerzielle CAD-Systeme unterstützen die Definition geometrischer Formen als Typen und deren Variationen. So ist es in den meisten Programmen möglich, ein Objekt zu erzeugen, abzuspeichern, und danach in geometrisch parametrisierter Form und mit Attributen versehen wieder einzusetzen. Dabei speichert das Programm nicht jedesmal ein neues Objekt einschließlich der Geometrie und Attribute ab; vielmehr greift es auf dessen Bezeichnung zurück und speichert nur die Abweichungen vom ursprünglichen Objekt. Beim Volumenmodellieren (Solid Modeling) verwendet man ein Set geometrischer Primitive, wie Quader, Keil, Kegel und Kugel, um mit diesen neue Objekte durch Boolesche Operationen herzustellen. Die Primitive haben dabei starken Typencharakter, denn sie sind durch Form und Attribute unverkennbar.

Der Grund für die Entwicklung des hier beschriebenen Instrumentes ergibt sich aus dem Bedarf, die Ordnung im Entwurf zu erhöhen, ohne zuvor umfangreiche planerische Aktionen zu starten, die mit dem Entwerfen

selbst nichts zu tun haben. Das Instrument soll in der Lage sein, ein beliebiges Element als Typ zu definieren, es mit Attributen zu versehen und ihm einen Namen zu geben. Ein Typ kann dabei ein Objekt oder die Repräsentation einer Funktion sein. Die Definition muß erstens vorausplanend möglich sein; dazu ist eine geeignete Definitionsumgebung notwendig, in der man einen Typ einfach und schnell generieren kann. Die Definition muß zweitens post facto möglich sein; dazu sollte man irgendein Objekt nachträglich als Typ definieren und mit den entsprechenden Attributen versehen können.

Sind Typen erst einmal definiert, müssen sie in der Modellierungsumgebung abrufbar sein. Eine vollständige, ikonenhafte Darstellung aller Typen ist notwendig, damit man Typen zum Einsetzen in das Modell auswählen kann. Außerdem muß das System die Typen nachträglich über ihre Attribute im Modell identifizieren können. In den seltensten Fällen wird jeder Teil eines Modells ein anderer Typ sein, denn es entspräche der Arbeit mit Einzelementen. Ziel ist die mehrmalige Wiederverwendung von Variationen eines Typs, um so ein generelles Ordnungsprinzip zu erreichen.

In der Vergangenheit war es ein Anliegen, Gebäudetypen zu definieren [Pevsner 1976]. Diese Kategorisierung ist interessant, da man sie sowohl zur Analyse als auch zur Synthese benutzt: Die meisten Gebäude lassen sich nachträglich einem bestimmten Gebäudetyp zuordnen, und beim Entwurf wird der Gebäudetyp oft als willkommener Rahmen gesetzt, innerhalb dessen sich das neue Objekt entwickelt. Ein Typ wird um so nützlicher, je öfter man ihn in ähnlicher oder abgewandelter Form im Entwurf einsetzen kann. Das bedeutet, daß die Manipulationsmöglichkeiten mehr als rein geometrischer Natur sein sollten. Komplexe Typen nähern sich daher der Definition von Prototypen an.



Instrument: Objektorientiertes Modellieren II - Substitution

Anwendung: Hierarchisches Denken

Verwandte Gebiete: Kreativität, Design

Substitution bedeutet das Ersetzen eines Objektes durch ein anderes. Das Prinzip der Substitution hat in unserem Fall drei Bedeutungen. Die erste betrifft das einfache Austauschen von Alternativen innerhalb der gegebenen Rahmenbedingungen, also das vollständige Ersetzen eines Objektes durch ein anderes. Die zweite Bedeutung betrifft das Ersetzen aller Objekte einer Klasse durch die entsprechenden Objekte einer anderen. Die dritte Bedeutung tangiert das wichtige Gebiet des Ersetzens von einfachen Platzhaltern oder Symbolen durch komplexe, realitätsnähere Objekte. In allen Fällen ist der syntaktische Vorgang gemeint, der eine, mehrere oder alle Variationen eines Typs durch andere austauscht.

Um den Vorgang der Substitution wiederholbar und kontrollierbar durchführen zu können, sind Konventionen notwendig. In der Substitution geometrischer Körper haben sich folgende Konventionen als nützlich erwiesen: Das zu ersetzende Objekt und das neue Objekt müssen kompatibel sein; das neue Objekt ist auf demselben Detaillierungsgrad wie das zu ersetzende einzusetzen; das neue Objekt soll den gleichen Referenz- oder Einsetzpunkt, sowie die gleiche Größe haben, um nach der Substitution ungewollte Verzerrungen zu vermeiden.

Substitution eignet sich gut für das Studium geometrischer Alternativen. Das Verfahren weist graphische Analogien mit einem Tabellenkalkulationsprogramm auf: durch das Ausführen von 'Was wäre, wenn!'-(What-If)-Simulationen ist der Effekt des Einsetzens neuer Alternativen sofort sichtbar. In einem sich entwickelnden Entwurf kann man mit der Definition einiger einfacher Typen beginnen. Beispiele sind geometrische Typen, funktionale Typen oder Kombinationen der beiden. Diese Typen enthalten Wissen oder sind der Ausdruck von Erfahrungen, die es wert

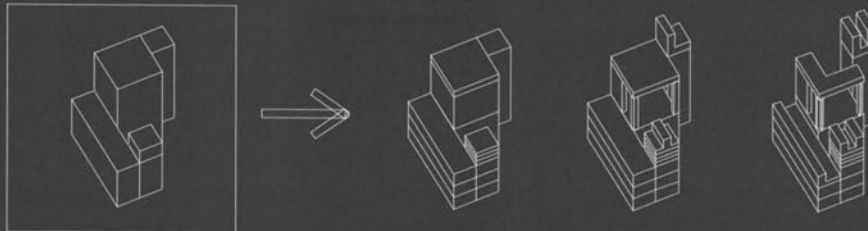
sind, als Typ definiert zu werden. Haben sich die Typen für Anwendungen bewährt, können sie in eine Bibliothek von Typen aufgenommen werden. Alle so erzeugten Eintragungen in der Bibliothek sind kompatibel und können durch Substitution gegeneinander ausgetauscht werden. Sobald ein Schema entstanden ist, können Teile desselben durch andere ersetzt und auf diese Art Alternativen untersucht werden.

Die gegenwärtigen Modellertechniken in CAD-Programmen lassen wenig Raum für undefiniertheit, wie sie bei Handzeichnungen oft auftritt. Für Computerprogramme ist es notwendig, genau zu wissen, welches die geometrischen Eigenschaften jedes einzelnen Entwurfselementes sind. Das Konzept eines substituierbaren Entwurfsvokabulars entwickelt sich ganz natürlich aus der Arbeit mit dem neuen Werkzeug. In diesem Fall ist es die Einzigartigkeit des Mittels, daß Entwurfende die zugrunde liegende Struktur ihres Entwurfs erkennen können, also auch solche Aspekte, die nicht notwendigerweise nur mit Geometrie in Beziehung stehen. Substitutions-Werkzeuge geben etwas mehr Freiraum, denn mit ihnen lassen sich aus anfänglich ungenauen, aber räumlich wichtigen Rahmenbedingungen präzisere Vorstellungen entwickeln. Substitution wird als Entwurfsinstrument dann fragwürdig, wenn ohne Nachdenken Entwurfsteile gegeneinander ausgetauscht werden. Dies mag für vorgefertigte Teile in einem fest definierten Raster angehen, aber kaum für formal und funktionell verknüpfte Architekturelemente. Solange die substituierten Objekte kein Wissen über sich selbst und über ihre Umwelt besitzen, werden sie jede noch so unsinnige Substitution mit sich geschehen lassen. Doch mit der Fähigkeit, Objekte mit Beschreibung von Wissen auszustatten, wächst die mögliche Eigenständigkeit der Objekte.

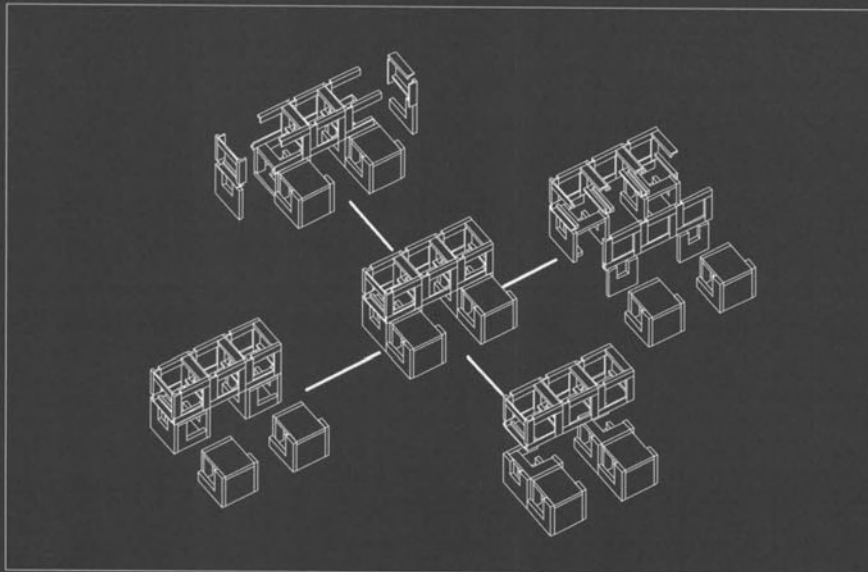
Substitution von
Buchstaben



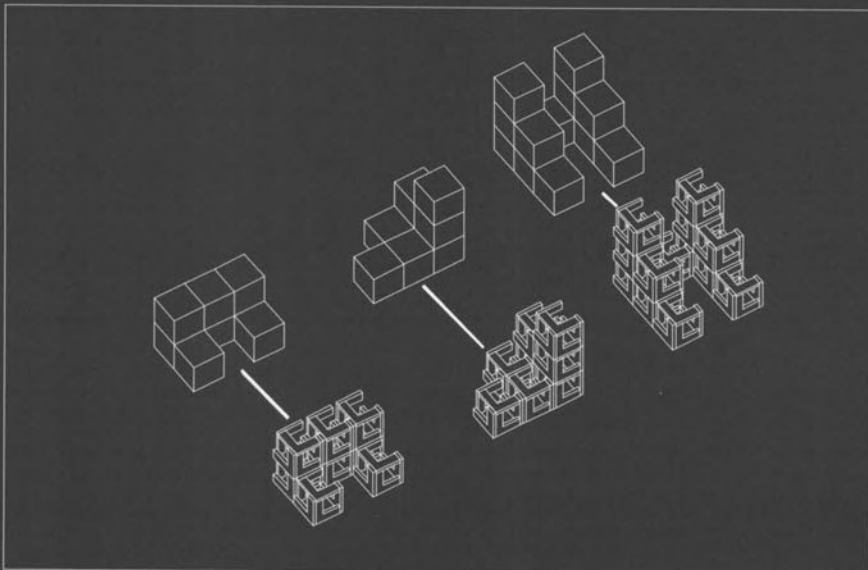
Substitution von
Formen



Vorstufe zur
Substitution:
Verschiedene
Interpretationen
einer Komposition



Substitution nach
Volumenstudien



Madrazo, 1993

Instrument: Objektorientiertes Modellieren III - Detaillierungsgrade

Anwendung: Entwurfsschritte, Entwurfsverfeinerung

Verwandte Gebiete: Antike Zeichnungen, Skizzieren

Der Detaillierungsgrad ist ein Instrument, um für den jeweiligen Stand des Entwurfs das richtige Maß an Objekt-Detaillierung zur Verfügung zu stellen. Er erlaubt, das gleiche Objekt in verschiedenen Detaillierungen zu betrachten. Essentielle Eigenschaften des Objektes, wie seine Position, Größe und Lage im Verhältnis zum Gebäude, ändern sich dabei nicht. Anstelle der Substitution durch einen anderen Typ geschieht hier der Ersatz durch einen anderen Detaillierungsgrad innerhalb desselben Typs. Gehören verschiedene Objekte zu einem Typ, so unterscheiden sie sich in der Regel nur durch ihren Detaillierungsgrad, also den Grad ihrer Ausarbeitung. Entscheidend ist dabei nicht die geometrische Form der Objekte, sondern ihre strukturelle Zusammengehörigkeit.

Das vollendete Bauwerk hat das Stadium der Abstraktion verlassen und den höchsten Detaillierungsgrad erreicht. Der Prozeß läßt sich auch umkehren: Vom Stadium der Vollendung können einzelne Aspekte wieder abstrahiert werden, um so eine Analyse zu ermöglichen. Das gleiche gilt für jedes Teil des Gebäudes. In der Zeit des Historismus schienen viele Gebäude nach außen einen hohen Detaillierungsgrad aufzuweisen: Ornamente bestimmten die Fassaden. Doch waren diese industriell in beliebiger Zahl herstellbar. In der Architektur Antoni Gaudis dagegen sind die Elemente mit hohem Detaillierungsgrad Unikate. In der darstellenden Architekturzeichnung wuchs der Detaillierungsgrad mit jedem Jahrhundert, um seinen Höhepunkt mit der Tradition der Beaux-Arts-Schule zu erreichen.

In der modernen Architektur ist der Detaillierungsgrad anscheinend verlorengegangen: Die auf dem Reißbrett entstandene Zeichnung und das architektonische Resultat haben sich für unnötig erachteter Ornamente und Schnörkel entäußert. Doch die innere Komplexität

der Gebäude wuchs. Außen- und Innenwände hochinstallierter, sogenannter intelligenter Gebäude (Intelligent Buildings) verbergen hinter der glatten Fassade ein äußerst komplexes Innenleben. Damit wird klar, daß der Detaillierungsgrad eines Gebäudes nicht von der sichtbaren Kleinteiligkeit in Fassade oder Grundriß abhängt, sondern daß auch die Funktionalität einen hohen Detaillierungsgrad aufweisen kann. Die Verwendung des Begriffes Detaillierungsgrad eignet sich demnach gut für architektonische Anwendungen, in denen die Typen stark durch ihre Semantik bestimmt sind: Ein Fenster ist ein eindeutiger Typ, ebenso eine Tür. Verschiedene Detaillierungsgrade werden an der Erkennbarkeit nichts Wesentliches ändern.

Das Instrument des Detaillierungsgrades ist geradezu ideal, die Informationstiefe eines Bauteils zu vergrößern, denn damit lassen sich alle Stadien, von der einfachen Handskizze über zweidimensionale Zeichnungen bis zu komplexen dreidimensionalen Objekten, verfolgen. Bedingung ist, daß diese Detaillierungsgrade eines Objektes einen gemeinsamen Einfügestpunkt haben. Ist dies der Fall, sind den Variationen keine Grenzen gesetzt. Die Variation des Detaillierungsgrades ist ein Beispiel für ein Instrument, das es vor dem Aufkommen der Computer in praktikabler Form nicht gab. Erst sehr schnelle Graphikmaschinen und die Bereitstellung einer entsprechenden Datenstruktur erlauben die Variation des Detaillierungsgrades als Entwurfsinstrument.

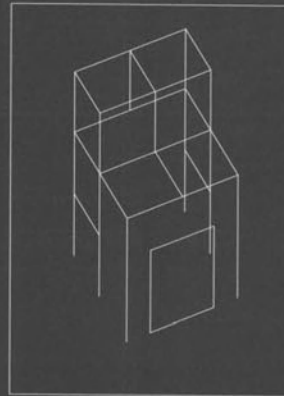
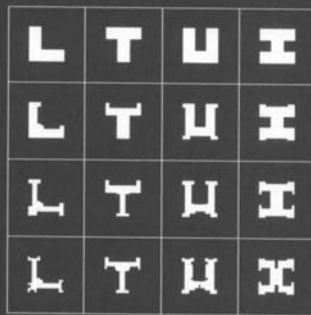
Das Instrument des Detaillierungsgrades führt den architektonischen Maßstab auf neue Art in das Computer Aided Design ein. Die Betrachtung eines Modells unter einem bestimmten Maßstab bedeutet nicht mehr Reduktion oder die Zunahme der geometrischen Komplexität, sondern kann jede Form einer anderen Sicht eines Modells annehmen.

Detailierungsgrad
in Buchstaben

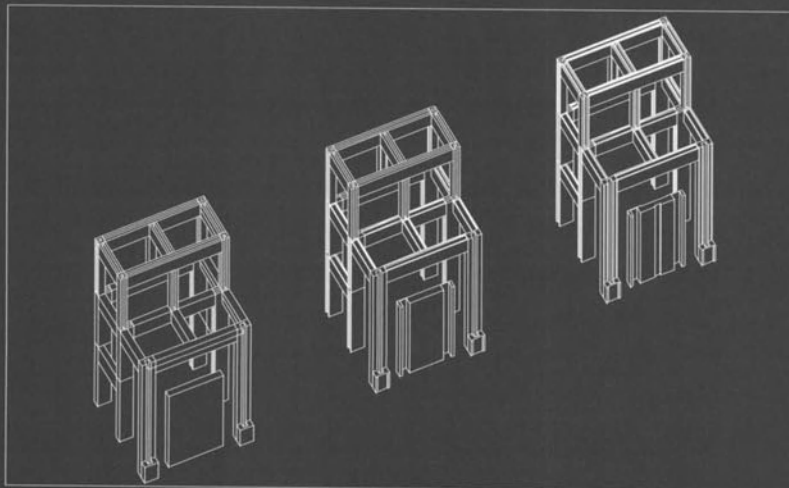


Detailierungsgrad
in Formen

Formales
Vokabular



Objekt



Mehrfache
Repräsentationen
desselben Objekts in
verschiedenen
Detailierungsgraden

Madrazo, 1993

Instrument: Objektorientiertes Modellieren IV - Designfokus - Logical Zoom

Anwendung: Entwurf

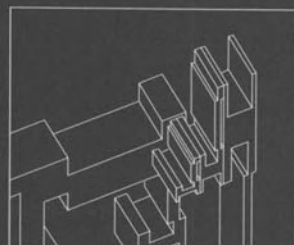
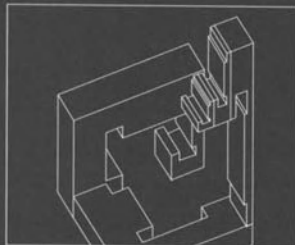
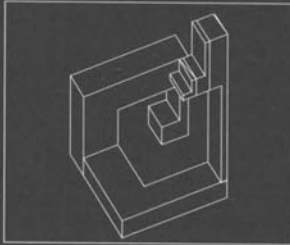
Verwandte Gebiete: Künstliche Intelligenz, Computergraphik

Das Prinzip des Designfokus oder des Logical Zoom ist in der Architektur von essentieller Bedeutung. Es erlaubt, in definierten Stufen von rein symbolischer Darstellung zu detaillierteren Repräsentationen überzugehen. Eine Parallele zu der traditionellen Planherstellung besteht im Detailgehalt der verschiedenen Maßstabsdarstellungen. So ist es selbstverständlich, daß ein Lageplan keine Informationen über Fensterdetails enthält, oder daß in der genauen Darstellung einer Tür keine Informationen über das umliegende Gelände auftreten. Selbst die leistungsfähigsten CAD-Systeme benötigen die Fähigkeit des Logical Zoom. Die Vorstellung, daß in einem Modell alle Informationen vom Kontext bis zum 1:1 Detail, einschließlich aller Attribute vorhanden sein müßten, überfordert jedes CAD-System. Vielmehr wird ein intelligentes CAD-System Verbindungen und Vernetzungen zwischen Geometrie, Symbolen, Details und Attributen auf einer höheren Ebene speichern und nur bei Bedarf zusammensetzen. Damit wird auch eine neue Klasse von Operationen auf abstrakterem und daher effizienterem Niveau möglich. Die zugrunde liegende Idee ist die der schrittweisen Verfeinerung eines Entwurfsansatzes und der schrittweisen Weiterentwicklung einzelner Entwurfsideen. Dies bedeutet den Einsatz der Top-Down-Methode. Das Bedürfnis, das Modell eines Gebäudes unter verschiedenen Gesichtspunkten zu betrachten und die Eigenschaften des Objektes nacheinander oder parallel zu studieren, ist ein wichtiges Anliegen für Analyse und Entwurf. Qualitativ ist dies in der menschlichen Vorstellung möglich, quantitativ durchführbar wird es erst durch die entsprechende Repräsentation der Objekte im Computer. Wenn wir uns ein Gebäude vorstellen, erscheint es in verschiedensten Darstellungsformen im Gedächtnis. Jeder Betrachtende sieht etwas anderes in einem Entwurf, setzt andere Schwerpunkte. Dies wird

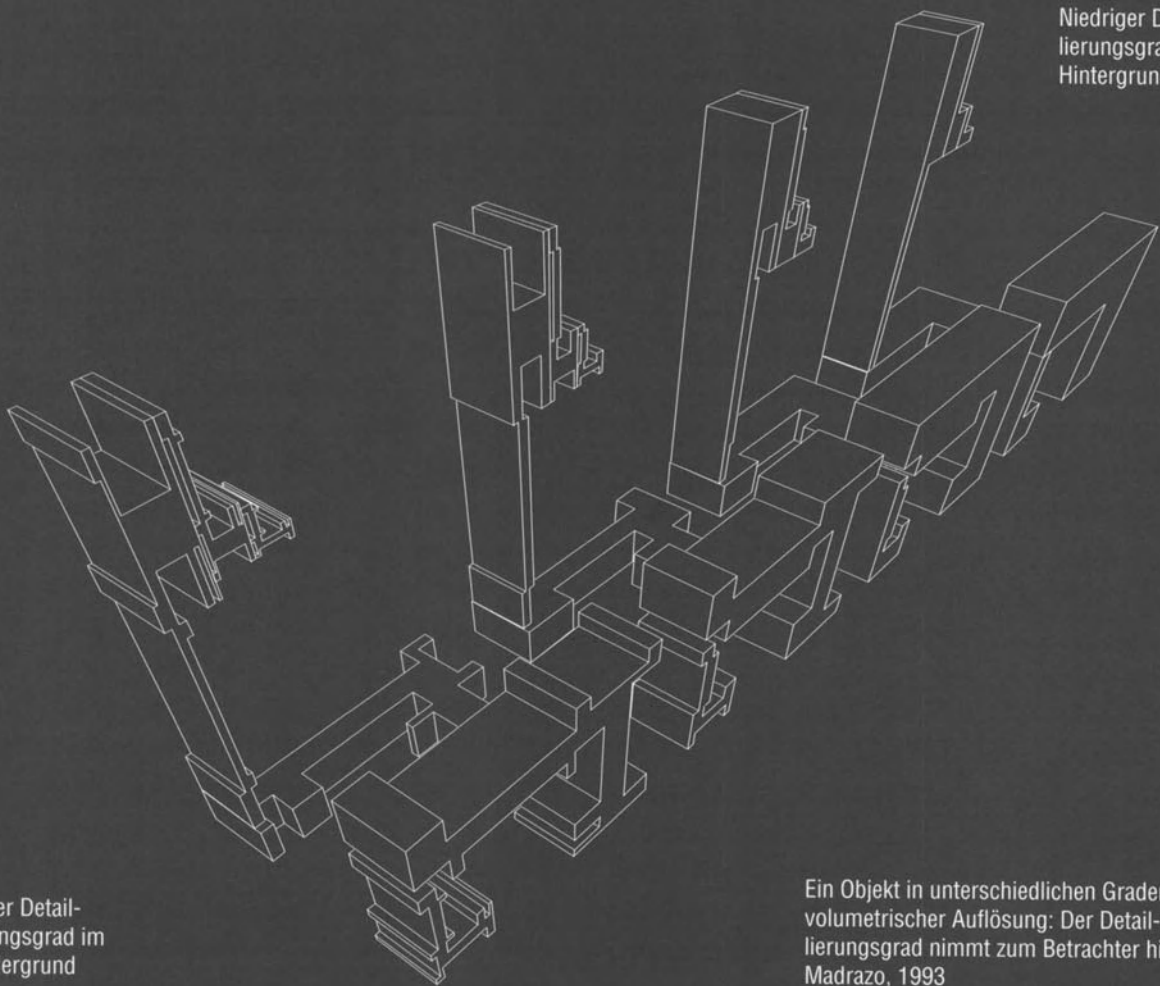
deutlich, wenn Architekten selbst zeichnen oder die Erinnerungen anderer in Bilder umsetzen, wie der amerikanische Künstler Douglas Cooper eindrücklich demonstriert [Cooper 1992].

Wir bezeichnen diese Fähigkeit als Logical Zoom [Wiesmann 1990], Fokus des Interesses [Galle 1992] oder Designfokus. Damit eröffnen sich vollkommen neue Sichtweisen eines Architekturmodells und des Entwerfens. Technisch läßt sich der Designfokus durch die konsequente Anwendung des objektorientierten Modellierens und dort insbesondere durch die Substitution und die Veränderung des Detaillierungsgrades erreichen. Doch selbst Architekturlehrer, die CAD im Entwerfen als Instrument einsetzen, haben das Potential des Designfokus noch wenig genutzt und orientieren sich hauptsächlich an den vorgegebenen Funktionen des eingesetzten CAD-Systems [Jacobs 1991]. In kommerziellen CAD-Systemen versucht man einen bestimmten Designfokus häufig mit Hilfe von Schichten (Layern) zu erzielen. Schichten sind jedoch ein zu schwerfälliges Instrument, um den Fokus des Interesses während des Entwerfens zu wechseln.

Es ist möglich, den Designfokus relativ einfach von der Zweidimensionalität auf die Dreidimensionalität zu verschieben, indem Bauteile für die entsprechenden Darstellung gegeneinander ausgetauscht werden. Kommerzielle CAD Programme bieten diese Möglichkeit vereinzelt an. Damit sind die Möglichkeiten des Designfokus aber nicht erschöpft. Die zuvor beschriebenen Abstraktionen und Simulationen zeigen ein Gebäude unter verschiedensten Sichtweisen. Der Designfokus erlaubt es, gleichzeitig, nacheinander oder in räumlich getrennter Abfolge alle formalisierbaren Aspekte eines Entwurfs oder bestehenden Gebäudes zu durchleuchten und sichtbar zu machen. Er wird dadurch zu einer neuen und attraktiven Modelliermethode.



Logical Zoom



Niedriger Detail-
lierungsgrad im
Hintergrund

Hoher Detail-
lierungsgrad im
Vordergrund

Ein Objekt in unterschiedlichen Graden von
volumetrischer Auflösung: Der Detail-
lierungsgrad nimmt zum Betrachter hin zu.
Madrazo, 1993

Zusammenfassung

Die Zahl bekannter, formalisierter und durch den Computer unterstützter Methoden und Instrumente wächst ständig. Noch werden die Methoden und Instrumente vorwiegend für die Analyse und Rekonstruktion eingesetzt, doch eine zunehmende Zahl junger Architektinnen und Architekten benutzt sie als selbstverständliche intellektuelle Instrumente für die Tätigkeit des praktischen Entwurfs. Immer öfter tauchen in Publikationen Entwürfe bekannter Architekten auf, bei denen zumindest ein Teil der Präsentation die Verwendung eines Computers erkennen läßt. Interessant ist die Tatsache, daß die Verwendung des neuen Instruments durch bekannte Architekten einen höheren Abstraktionsgrad aufweist als die Beispiele weniger bekannter Architekten, die den Realitätsgrad möglichst hoch ansetzen [Gandelsonas 1991]. Am besten läßt sich das Eindringen des CAAD in Lehre und Praxis in den letzten entscheidenden Jahren wohl am Beispiel des ACS-Preises bei der jährlich stattfindenden Wiesbadener ACS-Messe nachverfolgen. Von den ursprünglich 'braven' und konformen Darstellungen ist nach vier Jahren kaum mehr etwas geblieben. Christoph Zechner beschreibt die Phasen der Entwicklung so:

"Die erste Phase des computerunterstützten Entwurfs war gekennzeichnet durch die bloße Imitation bereits existierender Planungs- und Darstellungsmethoden. CAD-Programme simulierten das Zeichnen von zweidimensionalen Projektionen des Raumes und waren daher nichts anderes als teurere Reißschieben und Bleistifte.

Die zweite Generation der CAD-Programme ermöglichte die Erstellung eines dreidimensionalen Datensatzes im Rechner. Die Arbeitsweise, die daraus resultiert, pendelt zwischen Modellbau und Zeichnen und ist daher wesentlich leistungsfähiger als tradierte ana-

loge Verfahren. Die mathematische Organisation der Daten im Computer, die sich von Kantenmodellen über Flächenmodelle zu Volumenmodellen entwickelte, erzeugte jeweils charakteristische Darstellungsformen. So verfiel man bei Projektionen von Kantenmodellen dem graphischen Reiz von dichten Wire Frame Darstellungen und erlag ebenso der Perfektion, mit der Texturen, Schattierungen oder Reflexionen von Flächen- oder Volumenmodellen dargestellt wurden.

Mit der Steigerung der Rechengeschwindigkeit wurde in der dritten Generation die Kommunikation mit dem digitalen Modell verbessert. Interaktivität bedeutet sofortiges Feedback jeder Manipulation des Datensatzes und löst den bisher eher linearen Planungsprozeß immer mehr durch einen verschachtelten Prozeß ab. Animationen ersetzen Einzelbilder, der Faktor Zeit kann leichter illustriert werden.

Die vierte Generation der Programme ist vor allem durch die ergonomische Gestaltung der Kommunikationsperipherie gekennzeichnet. User-Interfaces, wie Data-Suit oder Stereo-Display-Helme ermöglichen das Einklinken in den Cyberspace, ein interaktiver Raumsynthesizer regelt und steuert dank intuitiv erfaßbarer Benutzeroberflächen den Bau von virtuellen Welten. Die Simulationen verselbständigen sich bereits in Sparten wie der Unterhaltungsindustrie, der virtuelle Raum löst den realen ab." [Zechner 1992].

Hätte es eines weiteren Nachweises der Beziehung zwischen Musik und Architektur bedurft - ein beliebtes Thema in Vergangenheit und Gegenwart [v. Maur, 1985, S. 408] - so wird sie durch den Begriff des Raumsynthesizers wirkungsvoll erbracht. Genauso wenig allerdings wie ein Synthesizer das Komponieren guter Musik garantiert, werden die neuen Architekturmethoden und Instrumente den Entwurf guter

Bauten garantieren. Doch wie in der Musik werden sich auch die fähigsten Architektinnen und Architekten der neuen Instrumente bedienen.

Eine weitere entscheidende Entwicklung hat jedoch bereits begonnen: Langsam und von den meisten unbemerkt verschieben die neuen Methoden und Instrumente unser Verhältnis zur Realität. In den frühen Tagen des CAAD schien es das Ziel zu sein, Werkzeuge für die Unterstützung altbekannter Tätigkeiten zu besitzen. Mit der Beherrschung der Werkzeuge wuchs die Erkenntnis, daß sich diese auch für weitere Tätigkeiten nutzen ließen oder neue Tätigkei-

ten erst erlaubten. Die Werkzeuge änderten so die Arbeitsweise der Anwenderinnen und Anwender. Die nächste Stufe war der Versuch, die gebaute oder zu bauende Realität so nah wie möglich zu imitieren, wozu neue Instrumente, besonders in der Geometrie und für die Visualisierung notwendig wurden. Die Beherrschung dieser Werkzeuge wiederum definiert neue Sichtweisen der Architektur, sei sie projiziert oder gebaut. Damit öffnete sich der Weg zu einer virtuellen Architektur, die beginnt, sich von dem zu lösen, was wir traditionell als Realität betrachten und die dieser eine neue, virtuelle Realität gegenüberstellt.

Et

Et

Anwendungen in Lehre, Praxis und Forschung

Der Raum zwischen Architectura - die wir als theoretisch fundiertes Gedankengebäude charakterisiert haben - und Machina, deren Beschreibung im letzten Kapitel erfolgt, enthält Beispiele, welche die Wechselwirkung von Architektur und Maschine erzeugt. Diese Beispiele sind wichtig, zeigen sie doch die Distanz zwischen dem mit der Maschine oder in der Theorie Denkbaren und dem praktisch Realisierbaren. Sie gewähren einen Einblick in das heute Mögliche und einen Ausblick in die Zukunft.

Das vorliegende Kapitel demonstriert auch die Schwierigkeiten bei der Interaktion zwischen Architektur und Maschine, also die Anwendung des Computers in der Architektur und im Entwurf. Eine der Schwierigkeiten ist die Berücksichtigung der Tatsache, daß eine wachsende Zahl von Anwenderinnen und Anwendern einen Computer bereits als Teil der normalen Infrastruktur vorfindet, also keine Möglichkeit hat, neue Hardware oder Software zu evaluieren und zu beschaffen. Diese Gruppe hat hauptsächlich das Bedürfnis, das Maximum aus der gegebenen Situation zu machen. Diese Situation wird dadurch verursacht, daß diejenigen, die über Hardware und Software entscheiden, in der Regel nicht die Anwender sind. Daher sind die an die Maschine herangetragenen Erwartungshaltungen sehr verschieden. Im Extremfall bewegen sich die Erwartungen zwischen Produktionssteigerung und größerer Kontrolle auf der einen und Kreativitätssteigerung und Schaffung neuer Freiräume auf der Anwenderseite.

Das Thema Anwendungen ist auch deshalb schwierig, da es brisante Konflikte beinhaltet. Einerseits lassen sich Gebiete identifizieren, auf denen die Maschine traditionelle Vorgehensweisen sehr gut unterstützt. Es sind dies die allgemeine Büroautomatisierung, das Zeichnen und das Berechnen. Man könnte es dabei belassen und den möglichst rationellen Einsatz des

Computers auf diesen Gebieten propagieren. Andererseits wäre der Einsatz des Computers nur für diese Zwecke eine Fehlinvestition, da die Fähigkeiten der neuen Technologie dadurch bei weitem nicht ausgenutzt werden. Schlimmer noch, die alleinige Abstimmung eines Programms auf etablierte Paradigmen kann jede Art von Fortschritt verhindern, wie dies bei einigen CAD-Programmen bereits sichtbar wird. In diesem Kapitel interessieren deshalb hauptsächlich innovative Anwendungen, welche die neuen Möglichkeiten der Computer untersuchen.

In der Lehre sind die Beispiele zugleich experimentell und praktisch. Nach dem Entwurf und der Implementation der Programme folgt deren praktische Anwendung im Unterricht. Der modulare Aufbau der Programme erlaubt den kontinuierlichen Ausbau des intellektuellen Instrumentariums. CAAD in der Lehre hat die Ausbildung zu kompetenten Architektinnen und Architekten zum Ziel.

In der Praxis sind CAD-Programme und mehr noch die zahlreichen Textverarbeitungs- und Tabellenkalkulationsprogramme inzwischen zum bewährten Hilfsmittel avanciert. Hier stehen die arbeitserleichternden und arbeitssparenden Fähigkeiten der Programme im Vordergrund, doch sind auch bereits die entwerferischen Aspekte der neuen Medien berücksichtigt.

In der Forschung sind die Maschine und die neu entwickelten Methoden und Instrumente zum unabdingbaren Werkzeug geworden. Sie sind zugleich Arbeits- und Testinstrumente. Der Ausblick am Ende des Kapitels gibt eine Beschreibung von heute noch exotisch erscheinenden Techniken, die allerdings punktuell in anderen Ländern, wie etwa in Japan, bereits in Gebrauch sind. Typisches Beispiel ist das Gebiet der Virtuellen Realität, das sich zu einem der interessantesten Forschungsbereiche entwickelt.

Lehre und Machina

Die Ergänzung architektonischer Kompetenz

In der Geschichte der Architekturausbildung ist es ungewöhnlich, Instrumente oder Methoden in der Lehre zu vermitteln, bevor sie in der Praxis ihren Nutzen bewiesen haben. Und trotzdem finden sich CAD oder CAAD inzwischen in beinahe allen Architekturschulen der industrialisierten Länder als Pflicht- oder Wahlfach. Es handelt sich dabei um einen kostenintensiven Lehrbereich, dessen gesamte Infrastruktur im Rhythmus weniger Jahre zu erneuern ist. Außerdem entwickelt sich die CAAD-Forschung mit großer Geschwindigkeit, was eine ständige Anpassung der Lehrinhalte notwendig macht. Zur Erklärung dieser an sich erstaunlichen Tatsache läßt sich anführen, daß CAAD wie andere Gebiete der Wissenschaft auf Prinzipien beruht, die einem weniger schnellen Wandel unterworfen sind. Die im ersten Kapitel beschriebenen Methoden und Instrumente sind Beispiele dafür. Verschieden sind nur die Vollständigkeit und die Art der Umsetzung der Prinzipien in den verschiedenen Programmen.

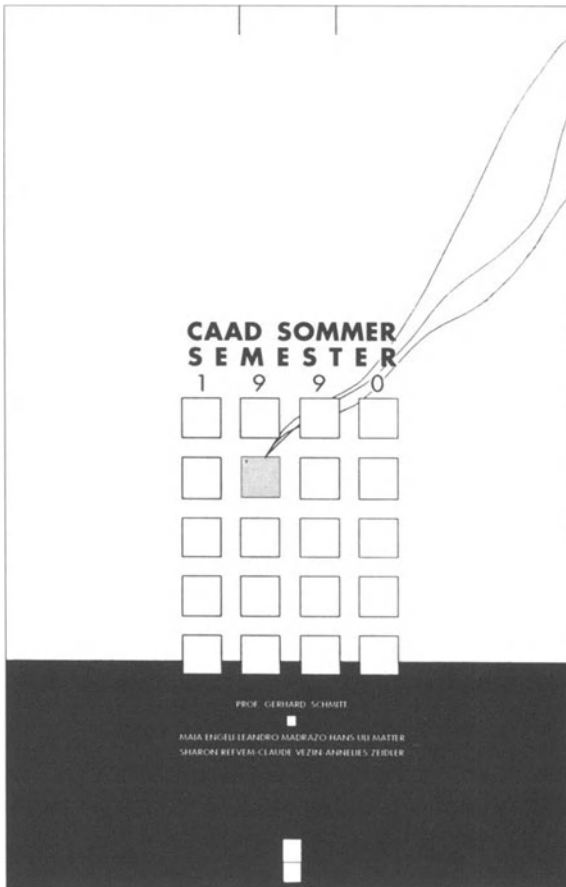
Die Vermittlung von Wissen über den Einsatz von Computern gehört heute, noch stärker aber in Zukunft, zur Grundausbildung von Architektinnen und Architekten. CAAD steht dabei selbstverständlich im Mittelpunkt. CAAD kann in Zukunft in der Ausbildung eine analoge Funktion wie der traditionelle Architekt im Entwurfs- und Bauprozess einnehmen - es kann als Vermittler und Koordinator dienen. Der computerunterstützte architektonische Entwurf wird somit in der Ausbildung nach und nach an die Stelle der traditionellen Entwurfslehre treten. Damit dies ohne negative Seiteneffekte geschehen kann, ist die allmähliche Einführung des neuen Instrumentariums notwendig, das den traditionellen Methoden dort den Vortritt läßt, wo diese effizienter sind. Zugleich darf man sich nicht scheuen, über-

kommene Lerninhalte und Lernmethoden durch die neuen Möglichkeiten zu ersetzen. Das Ziel muß dabei immer eine verantwortliche Architektur sein. Verantwortlichkeit bezieht sich nicht nur auf eines der wohl definierten technischen Gebiete, sondern im weitesten Sinne auf das Resultat: Architektur als kulturelles und gesellschaftliches Objekt.

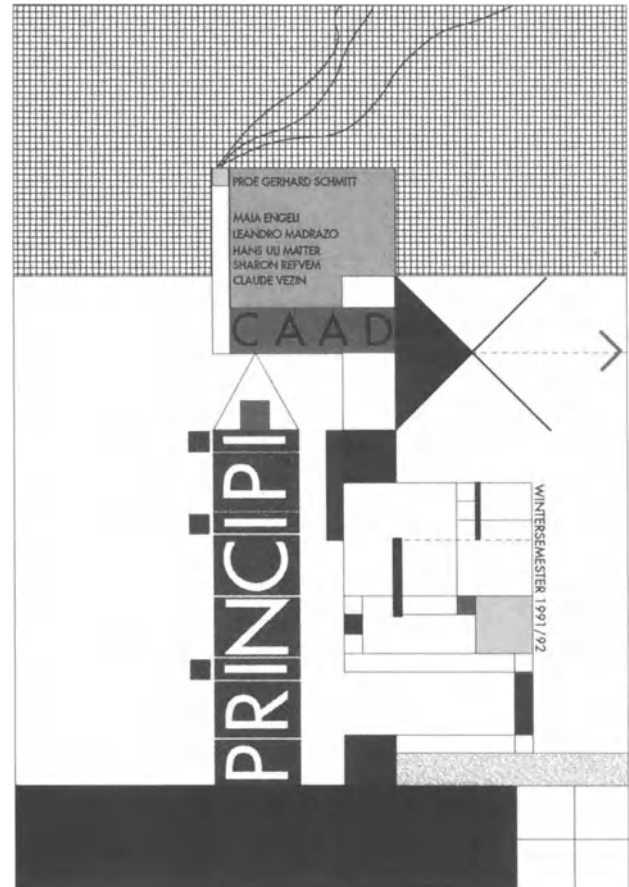
Wie in der Forschung, so lassen sich in der CAAD-Lehre intellektuelle Rahmen als nützliche Gedankenstützen verwenden. Und wie in der CAAD-Forschung gibt es auch in der CAAD-Lehre zumindest zwei komplementäre Richtungen. Die erste setzt Programme ein, um als nützlich erkannte Regeln und Vorgehensweisen zu unterstützen. Die zweite benutzt den Computer, um neue und für die Architektur vielversprechende Methoden und Instrumente zu entdecken.

Die folgenden Gedanken beziehen sich primär auf die Hochschule und basieren auf eineinhalb Jahrzehnte Erfahrungen im CAAD- und Entwurfsunterricht, sind jedoch in abgewandelter Form auch für andere Ausbildungsstätten anwendbar. Daraus ergeben sich verschiedene Forderungen für die CAAD-Lehre:

- (1) Die Vermittlung der Prinzipien. Damit erhalten die Studentinnen und Studenten einen Überblick über die vorhandenen Möglichkeiten und lernen, größere Zusammenhänge zu erkennen.
- (2) Die Vermittlung praktischer Beispiele. Die Prinzipien werden erst durch praktische und einfache Schulbeispiele anschaulich. Diese zeigen die Lösung kleiner, wohl definierte Probleme.
- (3) Die direkte Einbindung in den Entwurf. Es zeigt sich immer wieder, daß die Mehrheit der Studierenden am motiviertesten ist, wenn sie das Gelernte direkt im Entwurf oder einem anderen wichtigen Fach einsetzen können.



Semesterplakat Sommersemester 1990,
Madrazo, 1990



Semesterplakat Wintersemester 1991-92,
Madrazo, 1991

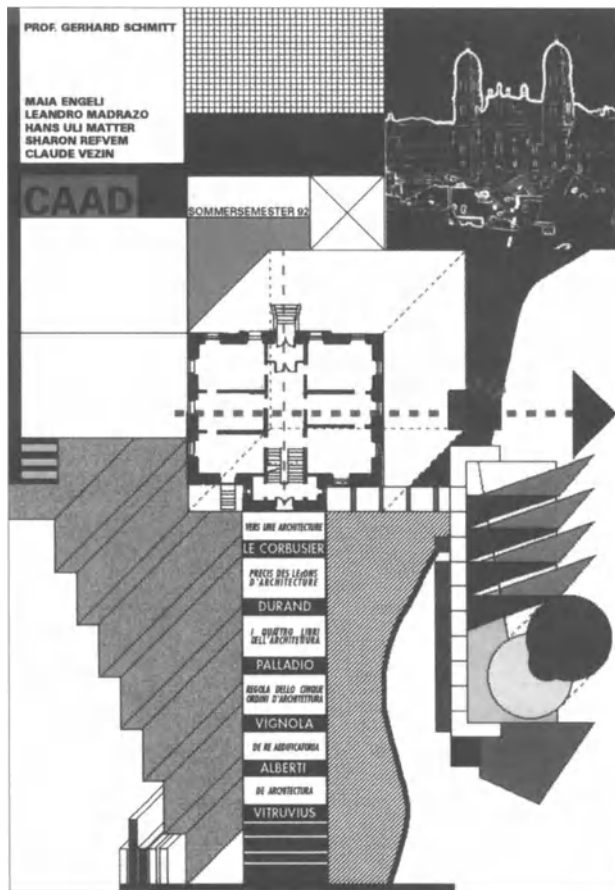
Die Lehre einer Hochschule, die eine dieser Komponenten ausläßt, ist nicht vollständig. Prinzipien ohne Beispiele sind im CAAD-Bereich schwer zu vermitteln und ziemlich nutzlos. Computereinsatz im Entwurf ohne Vermittlung der Prinzipien verleitet zu kritikloser Anwendung kommerzieller Programme. Dies führt zu einem im CAAD besonders gefährlichen Halbwissen, das beim Einsatz der nächsten Generation der CAAD-Programme versagen wird. Aus den Forderungen leiten sich die folgenden Lösungsmöglichkeiten ab:

(1) Einsatz eines CAAD-Programms, das alle Anforderungen erfüllt. Ein solches Programm existiert nicht und wird in absehbarer Zukunft nicht existieren, es sei denn, die Lehre paßt sich den Fähigkeiten der Software an.

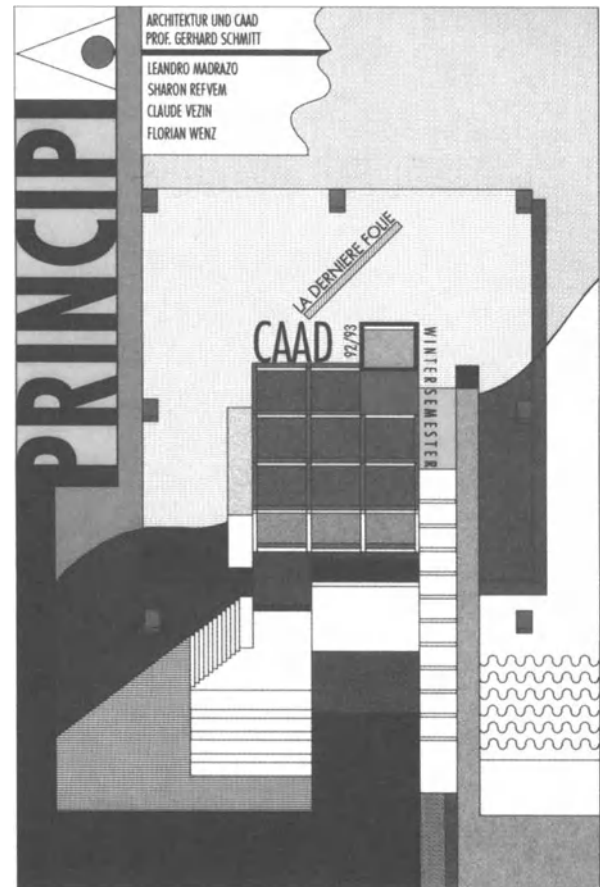
(2) Einsatz verschiedener Programme, mit denen jeweils ein Teil der Ausbildung geschieht. Für diese Lösung spricht, daß man dadurch die größte Übereinstimmung zwischen Programm und Anforderungsprofil erreicht, gegen sie spricht, daß das Erlernen eines neuen Programms jedesmal viel Zeit erfordert und sich damit der kurzfristige Lernerfolg reduziert.

(3) Anpassung des Lehrstoffs an die zur Verfügung stehenden Programme. Dies ist oft der Fall, wenn Zeitmangel und Beschränkungen der Infrastruktur keine andere Wahl lassen, und führt zu starken Vorgaben durch die verwendete Hard- und Software. Die Ergebnisse sind jedoch oft nicht die schlechtesten.

(4) Anpassung bestehender Programme an den Lehrstoff. Diese Möglichkeit ist die für die Lehrenden arbeits- und zeitintensivste, hat aber große Vorteile für



Semesterplakat Sommersemester 1992,
Madrazo 1992

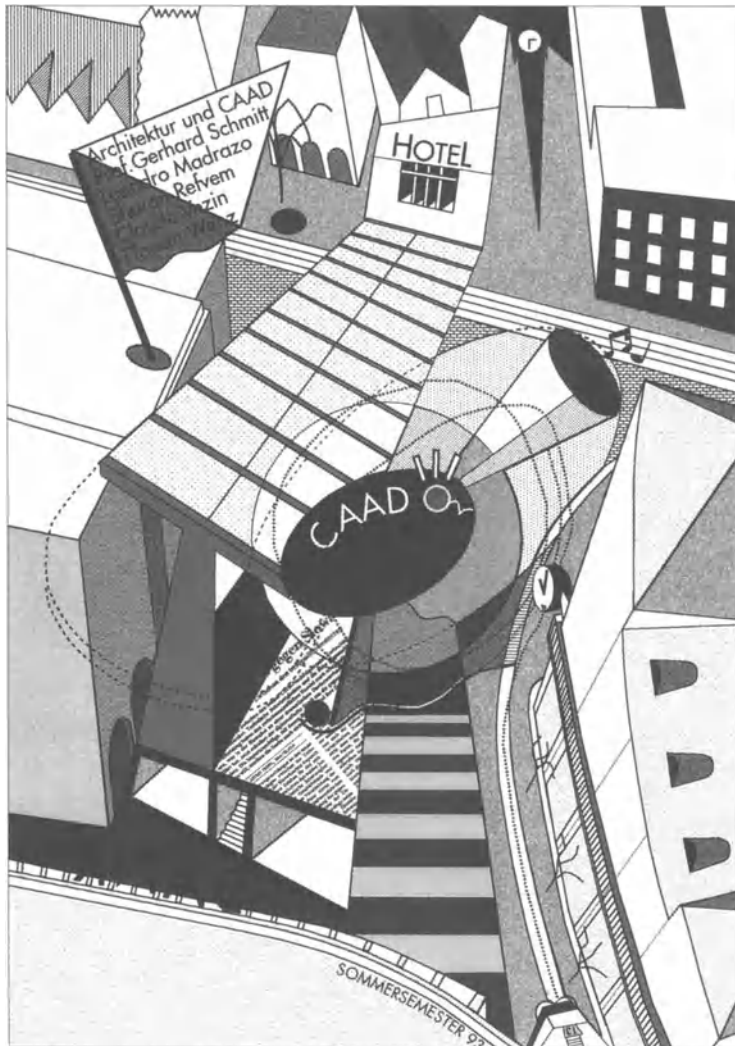


Semesterplakat Wintersemester 1992-93,
Madrazo 1992

die Studierenden. Zum einen kann so eine stabile kommerzielle Software zum Einsatz kommen, zum anderen lassen sich eigene Ideen in diese Software einbauen. Die meisten modernen CAD-Programme bieten zumindest die Möglichkeit einer Makroprogrammierung, mit der sich primitive Funktionen zu komplexen Kombinationen verknüpfen lassen. Dies ist die Option, welche wir gewählt haben.

Mehr als die Computerinfrastruktur sind das Wissen und die Einstellung der Lehrenden für den Erfolg von Bedeutung. Computer Aided Architectural Design ist keine Fähigkeit, die man völlig mühelos erlernt. Wenige - in der Regel die ohnehin begabten Architekturstudentinnen und Studenten - haben damit keine Schwierigkeiten. Die meisten kommen nach Anfangsproblemen zu den erwarteten Erfolgen. Und

eine dritte, kleine Gruppe kann trotz oder wegen größtem Einsatz zu keinem befriedigendem Ergebnis kommen. Trotz unserer Forderung, besonders die innovativen Aspekte der neuen Medien zu unterrichten, darf die Verwendung der neuen Technologie in der Lehre nicht vollständig Experimenten vorbehalten sein, sondern muß auch die traditionell gewachsenen und wichtigen Lehrinhalte unterstützen. In diesem Zusammenhang wird - wie immer bei der Einführung von neuen Methoden - die Frage diskutiert, bis zu welchem Grad das neue Arbeitsmittel die bestehenden Methoden imitieren und unterstützen soll, oder ob neue Methoden und Instrumente Inhalt der Lehre sein dürfen. Tatsache ist, daß auch in der Lehre das Verlassen der vorgegebenen Pfade zu Entdeckungen führen kann. Der Computer eignet sich hervor-



Semesterplakat Sommersemester 1993.
Thema: Pavillon auf dem Paradeplatz in
Zürich. Madrazo, 1993

gend dazu, durch Simulationen zuvor unbekannte Territorien zu erkunden. Die Verwendung des Computers wird zudem dadurch erleichtert, daß in der Lehre das Vermitteln von Prinzipien einen hohen Stellenwert besitzt, während in der Praxis das pragmatische Denken überlebenswichtig ist. Es ist notwendig, die Entwurfshilfen, die Computer bieten können, emotionslos zu untersuchen und kontrolliert damit zu experimentieren. Es gibt keinen vernünfti-

gen Grund, lediglich bestehende Methoden auf dem Computer zu implementieren. Ebenso wenig aber gibt es Anlaß, Entwurf nur mit neuen Methoden ausführen zu wollen. In diesem Sinn und mit dem Ziel einer intelligenten Entwurfsumgebung vor Augen implementieren wir seit Jahren neue Methoden und Instrumente für die Lehre. Einige davon und die Ergebnisse ihrer Anwendung sollen hier vorgestellt werden.

Das neue Instrumentarium

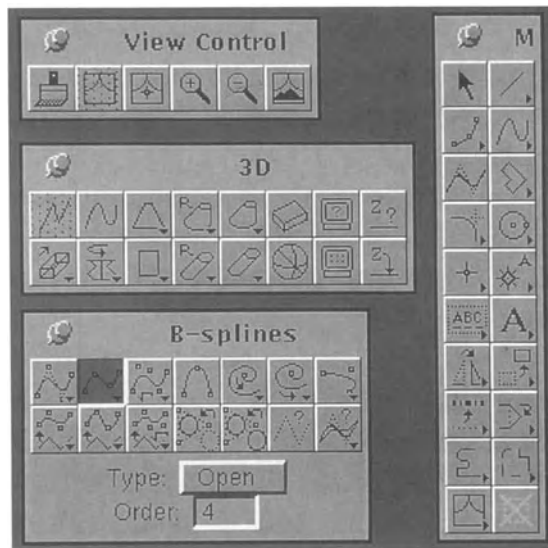
Die meisten der Studierenden haben bereits vor Studienbeginn in Schule und Ausbildung Kontakt mit einem Computer gehabt. Computerunterstützter architektonischer Entwurf dagegen ist eine Anwendung, die den Wenigsten bekannt ist. Wichtig ist der Zeitpunkt, zu dem die Studierenden mit dem neuen Instrumentarium in Berührung kommen: Geschieht dies zu Beginn des Studiums, so kann CAD zum selbstverständlichen Entwurfsinstrumentarium werden. Eine Folge kann sein, daß die entsprechenden Studentinnen und Studenten traditionelle Darstellungsformen nicht mehr erlernen. Geschieht die Einführung während oder gegen Ende des Studiums, so sind die wichtigsten Prägungen bereits geschehen. Ein mögliche Folge ist die falsche Einschätzung des neuen Instruments und dessen Reduktion auf die Funktion eines elektronischen Bleistifts. An der ETH Zürich sind seit Jahren beide Methoden implementiert. Besonders interessant sind die Ergebnisse des ersten Jahreskurses in Entwurf und Konstruktion bei Herbert Kramel. Innerhalb dieses Kurses gibt es eine große, über 200 Studentinnen und Studenten umfassende Gruppe, die nach traditioneller Art unterrichtet wird, sowie eine kleinere Gruppe von etwa 80 Studierenden, die von Beginn an den Computer für Entwurf und Darstellung einsetzen. Die Computergruppe wird nach dem Zufallsprinzip ausgewählt. Diese Studentinnen und Studenten arbeiten in Dreiergruppen. Nach Aussage der Lehrenden sind die Ergebnisse der Studierenden in der Computergruppe meßbar besser als die der anderen. Mit jedem Jahr werden die Unterschiede zwischen beiden Gruppen größer: Nicht nur haben die mit dem Computer Arbeitenden mehr Zeit zum Entwerfen, sie nutzen auch traditionelle Mittel, wie Modelle und Skizzen, in höherem Maß als die andere Gruppe, die durch darstellerische Aufgaben besonders gegen Ende des Semesters stark beansprucht

ist. Allerdings beklagen Professoren des zweiten Jahreskurses, daß die mit dem Computer vertrauten Architekturstudierenden das Zeichnen nicht beherrschten, eine an sich zu erwartende Konsequenz. Doch ist es jetzt noch so, daß im Lauf ihres Studiums beide Gruppen beide Vorgehensweisen erlernen.

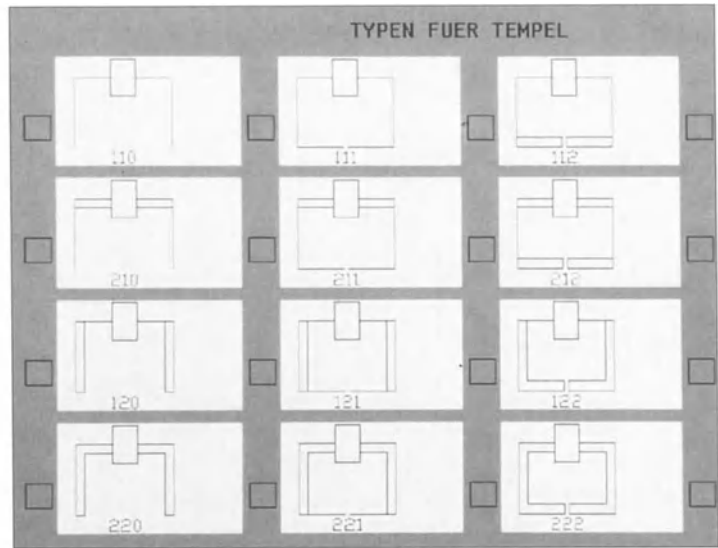
Selbst wenn der Einstieg bereits im ersten Semester erfolgt, stellt sich die Frage nach dem besten Weg der Einführung. Überspitzt lassen sich die Möglichkeiten auf zwei extreme Alternativen reduzieren: Entweder zunächst Vermittlung der Prinzipien und danach praktische Anwendungen, oder direkte Einführung am konkreten Beispiel und danach das Erläutern der Prinzipien. Die Lehrerfahrung zeigt, daß keine der beiden Alternativen, in letzter Konsequenz angewendet, zum Erfolg führt. Vielmehr liegt der erfolgreiche Weg wie meist in der Mitte.

Unbestritten ist die Bedeutung eines anfänglichen Überblicks über die Möglichkeiten und die tatsächlichen Anwendungen der neuen Technologie. Dies schließt die Textverarbeitung, die Tabellenkalkulation, Datenbanken und CAD-Programme vom Mal- und Zeichenprogramm bis zum integrierten Volumenmodellierer ein. Wichtig ist dabei, den Rahmen der bisherigen Anwendung und Forschung abzustrecken und den Lernenden die Möglichkeit zu geben, sich innerhalb der neuen Disziplin zu positionieren.

Den wenigsten Lehrenden und Lernenden ist das Potential zur Veränderung bewußt, welches das neue Instrumentarium bietet. Diese Veränderungen werden alle Bereiche des Architektenberufs und der Architekturausbildung erfassen. In der Ausbildung werden wir uns zunehmend auf die neuen Instrumente zu Lehrzwecken verlassen. Durch Simulationen werden die Studierenden das Verhalten und Aus-



Elektronische Instrumente: Palette eines CAD Programms, Intergraph Microstation, 1993



Intellektuelle Instrumente: Benutzeroberfläche für Tempelprototypen, Engeli, 1990

sehen ihrer Entwürfe zunehmend exakter vorherbestimmen können. In der Praxis wird durch die zunehmende Integration, die stärker werdenden Wünsche der Bauherrschaften nach Simulationen und durch die Veränderung in der Archivierung von Dokumenten mittelfristig ein starker Wechsel erfolgen.

Das neue Instrumentarium wird die Architekturausbildung wesentlich kapitalintensiver machen [Dette 1992]. An der ETH Zürich zum Beispiel war die Architekturabteilung 1992 mit 15% aller Studierenden die größte Lehreinheit. Doch bei den Ausgaben hielt sie sich mit 3,5% der Personalkosten und 2% der Sachkosten sehr zurück, trotz der Einführung von Computern in den Unterricht in den letzten Jahren. Traditionell kapitalintensive Departemente wie Mathematik und Physik dagegen verbrauchten für knapp

8% der Studierenden 16.5% der Personal- und 15.5% der Sachmittel [ETH 1993, S. 46]. Hier wird in der Zukunft eine Annäherung stattfinden.

Im Umgang mit dem neuen Instrumentarium ist von Seiten der Verwaltung und von Seiten der Studierenden ein Umdenken notwendig. Studentinnen und Studenten haben mit Hochleistungs-Workstations das preisliche Äquivalent von teuren Sportwagen mit hohen Wartungskosten auf dem Tisch, doch wegen des ähnlichen Aussehens werden sie behandelt wie PCs. Andererseits ist es wegen der besonderen graphischen Anforderungen der Architektur nicht sinnvoll, die Architekturarbeitsplätze mit den gleichen Standard-PCs wie die der textorientierten Studenten auszustatten. Die Ansammlung von Workstations in zentralisierten Clustern wird gegenüber dezentralen, vernetzten Arbeitsplätzen abnehmen.

Ordnung und Chaos

Es besteht die Ansicht, daß CAAD wie kein anderes Instrument in der Architektur zur Disziplin zwingt. Dies zeigen die vielen Leitfäden und Checklisten für den Einsatz von CAD im Bauwesen [Meißner 1992], in denen immer wieder ein diszipliniertes und geplantes Vorgehen empfohlen wird. Diese Anweisungen sind für den praktischen Einsatz von CAD in der Tat richtig. Sie sind jedoch auch ein Zeichen dafür, daß die meisten CAD-Programme noch nicht ausgereift sind [Dette 1992]. Je besser und intuitiver die Anwendung programmiert ist, desto mehr Freiheit besteht für die Benutzer. Ordnung und Chaos sind auch im Gebrauch von CAAD im Entwurf nahe miteinander verwandt. Chaos kann auf folgende Arten entstehen:

(1) Unkritische Übernahme bestehender Organisationsformen, und der Versuch ihrer direkten Spiegelung in Computerprogrammen. Dies ist seit der Einführung der ersten CAD-Programme oft geschehen. Zur Verfügung stehen jetzt Repräsentationsmethoden für Endprodukte und nicht Werkzeuge, die den Entwurf selbst unterstützen. Dies ist verständlich, waren doch die wenigsten Programmierer Architekten. Allen CAD-Programmen fehlt die Fähigkeit, dem dynamischen Fortschritt des Informationsgehalts während des Entwerfens Rechnung zu tragen. Die gegenwärtige Kommunikation und Organisation im Baubereich basiert auf dem Gebrauch externer Medien, wie Papier, Schrift und Zeichnung, sowie auf einer in Jahrhunderten gewachsenen Arbeitsteilung. Vieles, das sich zwischen den am Bau Beteiligten bewährt hat, läßt sich nicht direkt in eine Computerumgebung übertragen. Die Dynamik menschlicher Beziehungen, die Kapazität, Widersprüche aufzufangen und zur Zufriedenheit aller Beteiligten zu lösen, fehlt den Maschinen noch.

(2) In den frühen Entwurfsphasen eines Projekts sind zeitweilige Inkonsistenzen nicht unüblich, nicht ein-

mal unerwünscht [Mitchell 1990c]. Sie werden aber bei der Weiterbearbeitung und spätestens in der Ausführung zum kostspieligen und oft gefährlichen Ärgernis. Ein Beispiel ist die beliebte Cut-and-Paste-Methode. So werden zum Beispiel durch Scanning produzierte zweidimensionale Elemente gespeichert, ausgedruckt und auf die mit CAD erstellte Mutterpause aufgeklebt. Damit verliert das Dokument jegliche durch den Computer unterstützbare Konsistenz. Eine anscheinend schnelle Lösung ruft so bei Wiederholung immer größere Organisationsprobleme hervor und führt zur Unterhaltung von mindestens zwei inkompatiblen Datenbanken. Diese Tendenz wird sich fortsetzen mit jedem Teilbereich, der aus der Datenbank herausgelöst wird und sich verselbständigt.

Die Auswege aus dem Chaos sind mit herkömmlichen Methoden nicht begehbar. Es ist zwar mit bestehender Technologie möglich, ein Gebäude einschließlich allen relevanten Daten auf einer einzigen Compact Disk unterzubringen. Das Problem liegt nicht in der Technik, sondern in der Form, in der bauspezifische Information gewonnen, organisiert, gespeichert und abgerufen wird. Hier öffnet sich ein interessantes Gebiet für die Architekturforschung.

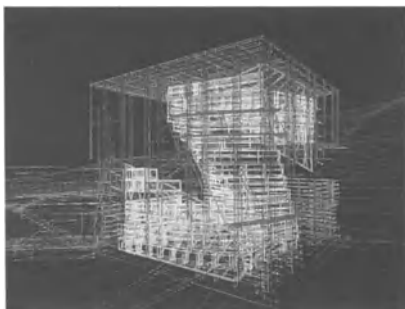
Kurzfristig ist das Chaos vermeidbar, indem nicht wahllos mit der Mischung von traditionellen und Computermethoden begonnen, sondern ein Weg für die stufenweise erfolgende Integration aller Daten in elektronischer Form festgelegt wird. Dies geschieht durch die Wahl kompatibler Software und Hardware, die eine spätere Integration oder zumindest Umwandlung in eine integrierte Umgebung nicht ausschließt. Integration kann klein beginnen - zum Beispiel durch das Verknüpfen eines CAD-Programms mit einem Tabellenkalkulationsprogramm, und die anschließend erfolgende Ablage in einer relationalen Datenbank.

Mittelfristig muß die Entwicklung weg vom Papier als einzigem Datenträger, hin zu den bereits existierenden Speichermethoden in Form von objektorientierten und intelligenten Datenbanken führen. Ein Großteil der Gebäudedaten ist schon in elektronischer Form, aber in völlig chaotischer Organisation vorhanden: Beschreibungen in Textverarbeitungsprogrammen, zwei- und dreidimensionale Modelle in CAD-Programmen, Rechnungen in Tabellenkalkulationsprogrammen, Details als Bitmap-Information. Eine wachsende und für alle am Bau Beteiligten sinnvolle Verknüpfung dieser Informationen wird positive Folgen haben.

Langfristig muß dem Computer überlassen werden, wozu er zumindest im Augenblick am besten geeignet ist: zur Organisation und Verwaltung von Daten. In nicht zu ferner Zukunft wird jedem Gebäude eine umfangreiche Datenbank zugeordnet sein, in der die Gebäudegeometrie, alle Verhandlungsunterlagen, Abrechnungen, Reparaturen, Energieverbrauchswerte und andere Merkmale gespeichert und intelligent abrufbar sind. Dafür müssen kommerzielle Daten-

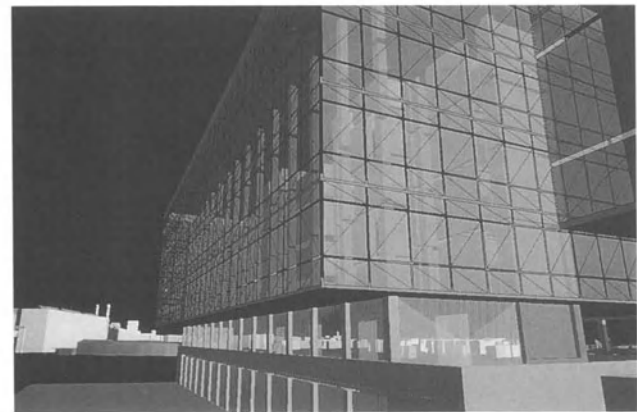
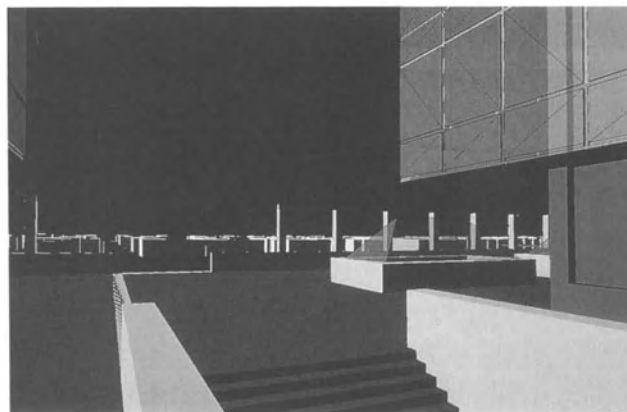
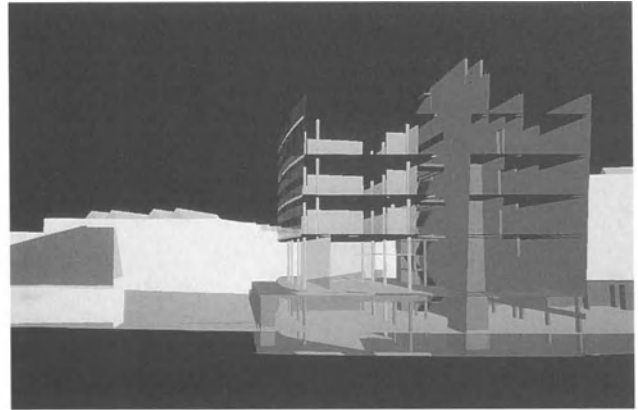
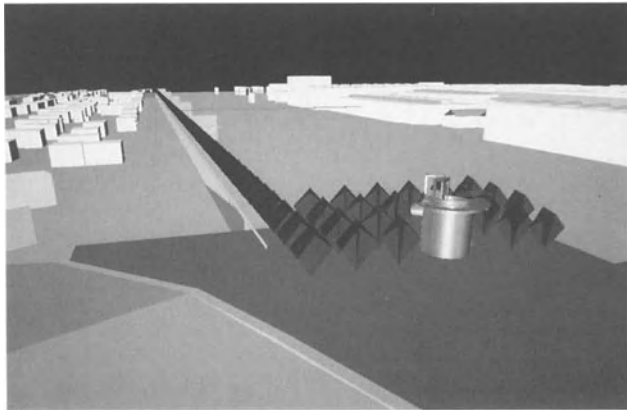
bank-Management-Systeme (DBMS) zur Verfügung stehen, die Informationen aller Art objektorientiert verwalten können. So werden auch die positiven Aspekte der zuvor kritisierten Cut-and-Paste-Methode nutzbar. Die CAD-Datenbank wird sich zwischen dem Entwurf, Ausführung und späterem Gebäude-Management in Charakter und Zweck leicht verändern: Sind während des Entwerfens dynamische Aspekte wichtig, so wird nach Bauabschluß die sichere Verwaltung großer Datenmengen an Bedeutung zunehmen. Nur so lassen sich aus gebauter Architektur wichtiges Feedback und Erkenntnisse für zukünftige Entwürfe gewinnen, und nur so kann im Laufe der Zeit ein nachvollziehbares Wissen im Baubereich aufgebaut werden.

Diese für die Praxis wichtigen Aspekte von Chaos und Ordnung müssen bereits in der Lehre berücksichtigt werden. Dazu ist es mittelfristig notwendig, die fehlende semantische Organisationsfähigkeit der CAD-Programme durch eigene Disziplin auszugleichen. Langfristig aber werden es die Programme übernehmen, Verknüpfungen herzustellen und zu unterhalten.



Digitales Modell einer Bibliothek für die Speicherung von traditionellen Dokumenten in einer unkonventionellen Form. Siehe auch Principia: Objektorientiertes Entwerfen. Sladoljev, 1992

Entwurfshilfen

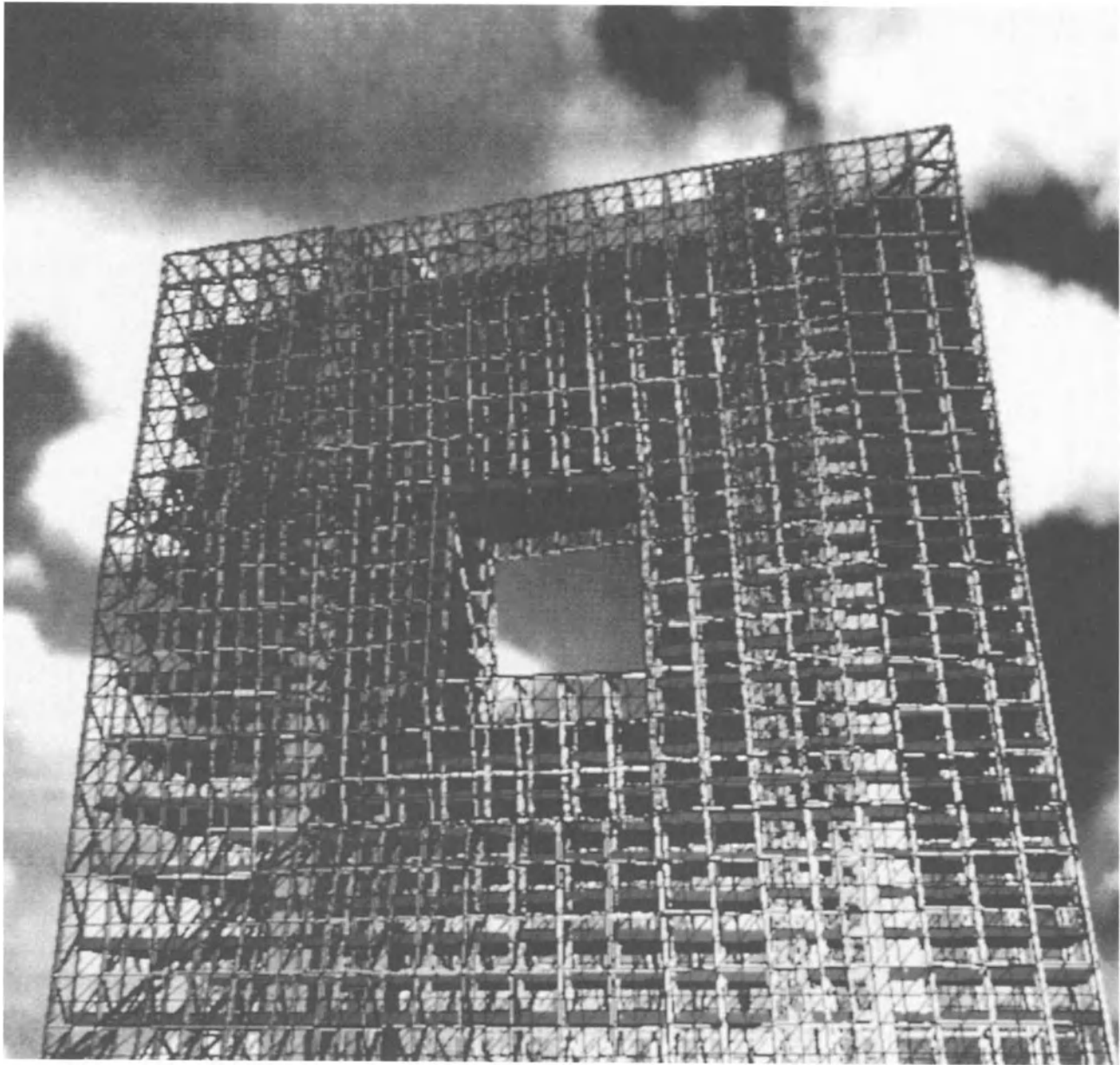


Das urbane Projekt Winterthur, Modellskizzen für die Zwischenkritik eines Entwurfs bei Mario Campi. Oben: Belvedere und Museum, Kämpfer, 1991. Unten: Verwaltungsbau auf dem Sulzer-Areal, Hirschberg, 1991

Bei der Einführung von Computern in den Entwurf gibt es die verschiedensten Ansätze, von denen hier lediglich zwei Extrempositionen dargestellt werden sollen. Das erste Extrem ist die Computerisierung bestehender Methoden, charakterisiert durch die Idee des elektronischen Bleistifts. Nach dieser Methode kommen nur solche Computer und Programme zum Einsatz, welche die bestehende Lehre- oder Praxis nachahmen. Experimente und Programme, die solche unterstützen, sind unerwünscht. Das zweite Extrem ist der völlige Bruch mit der bestehenden Entwurfs- und

Lehrpraxis. Programminstrumente werden um ihrer selbst willen eingesetzt, um die neuen Möglichkeiten zu erkunden, ohne Rücksicht auf überkommene Lehrvorstellungen.

Beide Positionen können bei entsprechendem architektonischem Wissen der Lehrenden zum Erfolg führen. Ist eine eigene Vorstellung eindeutig und überzeugend formuliert, so ist der Computer ein selbstverständliches Hilfsmittel, diese Vorstellung noch weiter zu verstärken. Auf der anderen Seite ist die Freigabe des Computers zum Experimentieren eine



Wettbewerb für ein Trade-Center in Zagreb. Das Experimentieren mit verschiedenen Programmen führte zur Klärung der architektonischen Idee. AutoCAD und Intergraph Microstation auf Sparcstation. Sladoljev, 1993

vielversprechende Option für schöpferische Studentinnen und Studenten, die sich ungern in vorgegebene Denk- und Arbeitsschemen einzwängen lassen.

Die Rolle des Computers als Entwurfshilfe wird sich auch im Laufe der Ausbildung ändern. Stehen zu Beginn des Studiums die lernintensiven Applikationen im Vordergrund, so werden gegen Ende die Möglichkeiten dominieren, mit der Maschine Verbindungen zu definieren und neue Erkenntnisse aus Bestehen-

dem zu gewinnen. Auch die generativen Fähigkeiten bestimmter Programme sind nicht zu unterschätzen. Interessant sind auch die Methoden der Prototypen, des fallbasierten Schließens sowie die Instrumente der Formengrammatiken, zu denen Beispiele folgen werden. So wie heute neben dem Instrument Zeichnung das physische Modell noch immer eingesetzt wird, so werden auch in der Zukunft die traditionellen Entwurfshilfen ihre Rolle im Entwurfsprozeß besitzen.

Integrierter Entwurf

Als integrierten Entwurf in der Architekturausbildung wollen wir vereinfacht die interaktive Einbeziehung von verschiedenen Architekturaspekten bezeichnen. Voraussetzung dazu ist ein Set kompatibler Abstraktionen. Alle so entstehenden Simulationsmöglichkeiten basieren auf einem digitalen Modell des Gebäudes. Die große Informationstiefe, die das Arbeiten mit Computerprogrammen erlaubt, macht den integrierten Entwurf erst möglich. Die zugrunde liegende Idee erscheint logisch: Alle meßbaren Größen werden während des Entwerfens simuliert und geben ein direktes Feedback, aufgrund dessen der Entwurf sich in eine bestimmte Richtung bewegt.

Trotz oder vielleicht wegen der Logik eines solchen Ansatzes ergeben sich in der Lehre erhebliche Schwierigkeiten der Verwirklichung, wie wir an verschiedenen praktischen Versuchen erfahren haben. Für die Überprüfung der Möglichkeiten des integrierten Entwurfs entwickelten wir 1986 ein einfaches Programm, das während des Modellierens Berechnungen für den Heiz- und Kühlbedarf, die Elektrizitätskosten, den Fensteranteil und die Erstellungskosten eines Gebäudes anstellte und diese in Zahlen und in Form eines Diagramms wiedergab [Schmitt 1988a, S. 157]. Obwohl die Ergebnisse der Berechnungen auf dem Bildschirm direkt neben dem geometrischen Resultat zur Verfügung standen, war bei der Entscheidungsfindung das Arbeiten mit dem geometrischen Modell entscheidend.

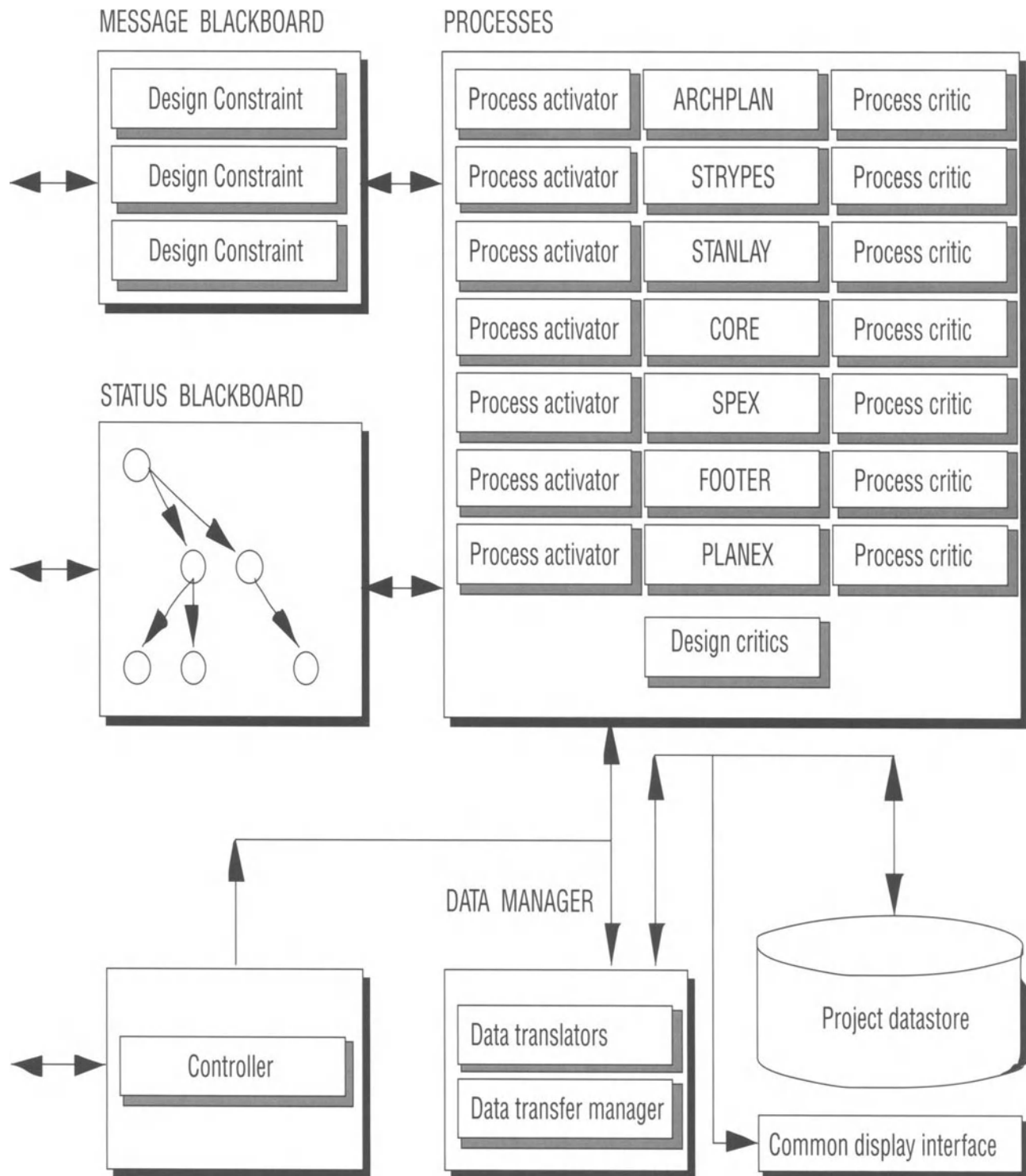
Bisher fehlt allen CAD-Programmen die Fähigkeit, dem dynamischen Fortschritt des Informationsgehalts während des Entwerfens Rechnung zu tragen. Zur Verfügung stehen heute Repräsentationsmethoden für Endprodukte und nicht Werkzeuge, die den Entwurf selbst unterstützen. Zur Verbesserung dieser Situation schloß sich eine Gruppe von Architekten und Ingenieuren an der Carnegie Mellon University in Pittsburgh

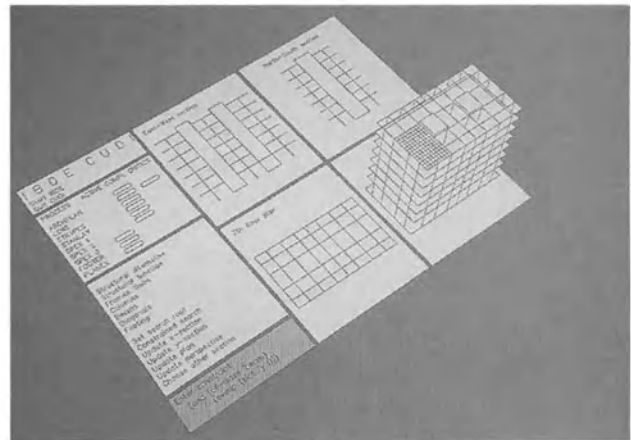
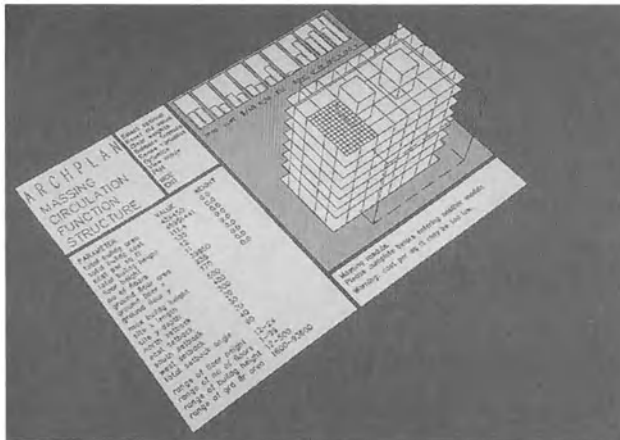
zusammen und begann die Arbeit an einer 'Integrierten Gebäude-Entwurfsumgebung' (Integrated Building Design Environment, IBDE). Im IBDE-Projekt [Fenves 1990] bilden insgesamt sieben wissensbasierte Programme zusammen mit einer integrierten, relationalen Datenbank ein System, das für die Unterstützung des Entwurfs von Bürohochhäusern gedacht ist. Die Beschränkung auf diesen Gebäudetyp ergab sich aus dem bereits bestehenden Wissen der Mehrheit der Projektteilnehmer. Die Programme liefen auf Workstations unterschiedlicher Hersteller und verwendeten verschiedene Computersprachen. Allen gemeinsam war der Anschluß an ein Netzwerk, über das sie mit dem Blackboard und der integrierten Datenbank in Verbindung standen, sowie eine Benutzeroberfläche, welche die Ergebnisse der einzelnen Prozesse und den neuesten Stand des Entwurfs zeigt. IBDE bestand aus den folgenden Komponenten:

ARCHPLAN repräsentiert Gebäude als Prototypen, die schrittweise verfeinert und den Forderungen der Bauherren angepaßt werden. Es ist in Lisp geschrieben und folgt der objektorientierten Programmierauslegung. Es nutzt verschiedene Formen des Schließens, wie Forward-Chaining, Backward-Chaining und Hill-Climbing für das Auffinden lokal bester Lösungen [Schmitt 1989b].

CORE besteht aus einem Prä-Prozessor, der den Input von *ARCHPLAN* entsprechend vorbereitet, einem Generator, der Layouts für Aufzugslobbies erstellt, sowie aus einem Post-Prozessor, der die besten Alternativen auf dem Bildschirm darstellt. Der Generator arbeitet mit einer Graphenrepräsentation der räumlichen Beziehungen und erzeugt mit einem modifizierten Branch-and-Bound-Algorithmus ein Set aller akzeptablen Lösungen. Testregeln ordnen jeder Alternative eine Note zu, die ausdrückt, in welchem Maße die Lösung

Architektur der integrierten IBDE
 Umgebung. Prozesse, Blackboards,
 Daten-Manager und Projekt-Datenbank
 sind in einer verteilten Computer-
 Umgebung vernetzt.





Zwei Vorschläge für eine neue, dreidimensionale und interaktive Benutzeroberfläche für ein integriertes Programm, basierend auf ARCHPLAN. Links Massenmodell-Modul, rechts Statik-Modul. Schmitt, 1989

nicht explizit erklärt werden. Entwerferspezifisches Wissen besteht aus Faustregeln, die dem Design eine bestimmte Richtung geben.

FOOTER zerlegt den Entwurf der Fundamente in Teilbereiche: Wahl eines Fundamenttyps, des Materials, der Erstellungsform, der Aushubform, sowie eines parametrisierten Fundamentdesigns. *FOOTER* verläßt sich stark auf Rahmenbedingungen und heuristische Regeln, um über die Relevanz eines bestimmten Designs zu entscheiden und um inkonsistente Alternativen auszuschließen.

PLANEX unterstützt die Bauablaufsplanung. Es beruht auf einer Reihe von Wissensquellen in Form von Entscheidungstabellen, die Sets von Regeln enthalten. Das Wissen wird jeweils auf ein Teilproblem der Gesamtplanung angewandt. Entscheidungstabellen finden auch bei der Wissensakquisition Verwendung [Hendrickson 1987].

Das System funktionierte folgendermaßen: Der Architekt erstellte interaktiv mit *ARCHPLAN* einen vorläufigen Entwurf des Gebäudes und schickte eine Repräsentation des Ergebnisses in Form von Frames oder Rah-

men an die integrierte Datenbank. Diese speicherte das Ergebnis und schickte die Meldung über einen zur Verfügung stehenden Entwurf an das Blackboard. Daraufhin wurden die anderen Programme aktiv und verlangten Input für ihre Prozesse. Ein der integrierten Datenbank vorgeschalteter Übersetzer formte die Daten in die für die individuellen Prozesse lesbare Form um. Sobald einer dieser Prozesse abgeschlossen war, konnten weitere Prozesse aktiv werden. Nach einer Rechenzeit von etwa zwei bis drei Stunden stand die vollständige Beschreibung eines Gebäudes zur Verfügung, die typischerweise eine Größenordnung von etwa 25'000 Einzelpositionen umfaßte. Anders als in den zuvor beschriebenen Beispielen war die Interaktion auf alphanumerische Eingaben beschränkt, während das Resultat in graphischer Form erschien. Die Schwierigkeiten in der praktischen Anwendung zeigten sich darin, daß die Nutzer das Gefühl hatten, ihnen seien viele wichtigen Entscheidungen abgenommen. Trotzdem war *IBDE* ein wichtiger Beitrag, um die Machbarkeit der Idee des integrierten Entwerfens realitätsnah zu testen.

Modelle und Präsentation

Traditionelle Architekturmodelle haben ihren festen Platz in der Entwicklung und Präsentation von Entwürfen in Lehre und Praxis. Das traditionelle Architekturmodell zeigt primär die formal-geometrischen Eigenschaften eines Entwurfs. Arbeitsmodelle erlauben ein interaktives und schnelles Testen von räumlichen Alternativen und Materialien. Präsentationsmodelle sind trotz der notwendigen Abstraktion ein Versuch der Vorwegnahme von Realität und dienen als Entscheidungshilfe. Im traditionellen Modell ist es möglich, den Modellelementen nur eine begrenzte Zahl von Attributen wie Farbe und Text zuzuordnen. Die sich daraus ableitenden Folgerungen müssen im menschlichen Gedächtnis analysiert werden. Im Unterschied zum physischen Modell gestattet es die Repräsentation im Computer, jedem Objekt eine Vielzahl von Attributen zuzuordnen, nach denen sich das Gesamtobjekt später analysieren läßt. Zusätzlich kann die zeitliche und inhaltliche Struktur des Entwurfsprozesses im Computermodell vorhanden sein. So ist es möglich, während des Entwerfens verschiedene Fixpunkte zu setzen, zu denen im Verlauf der Arbeit zurückgekehrt werden kann und die als Ausgangspunkt neuer Explorationen dienen können.

Aus diesen Beschreibungen wird deutlich, daß ein direkter Vergleich von traditionellen und von Computermodellen wenig sinnvoll ist. Die Wiederverwendbarkeit der traditionellen Modelle ist geringer als die der elektronischen. Im fortschrittlichen Entwurfs- und Konstruktionsunterricht stehen beide Arten von Modellen zur Verfügung und werden ihren Stärken entsprechend genutzt.

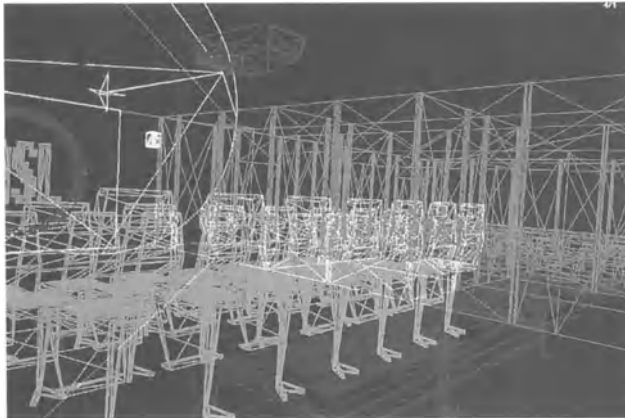
Die traditionelle Präsentation nutzt das Medium Papier als Träger aller wichtigen Informationen. Abstraktionen wie orthogonale und perspektivische

Projektionen sind das allgemein anerkannte Mittel, komplexe Gebäudeentwürfe allen an Entwurf und Ausführung Beteiligten zu verdeutlichen. Künstlerischen Weiterentwicklungen der Darstellung in der Entwurfsphase sind dabei keine Grenzen gesetzt. In der Dokumentationsphase werden die Bestimmungen über die Darstellung allerdings enger und formalisierter.

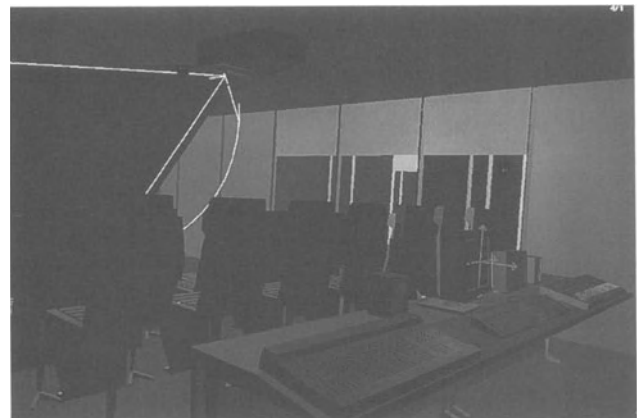
Die Präsentation mit dem Computer nahm die am meisten formalisierten Darstellungsweisen als Ausgangspunkt und hat dort in Form zahlreicher CAD-Programme, die Zeichen- und Konstruktionshilfe leisten, ihre professionellste Entwicklung erreicht. Obgleich es trivial erscheint, aus dreidimensionalen Computermodellen orthogonale Projektionen zu entwickeln, löst bisher kein Programm diese Aufgabe perfekt. Der Grund ist der hohe Grad an Spezialisierung und der Gehalt an Semantik, den die traditionelle Darstellung erreicht hat. Deshalb muß entweder der Dokumentationsstandard den Fähigkeiten des neues Mediums angepaßt, oder die dokumentenspezifische Extraktion aus einem multidimensionalen Modell vorgenommen werden. Die letztere Möglichkeit ist anzustreben, obwohl sie die Speicherung zusätzlicher Inhalte im Computermodell erfordert, die keine direkte Bedeutung für das geometrische Modell haben.

Visualisierungsprogramme beruhen auf Forschungsergebnissen in Gebieten wie etwa der Physik, welche beispielsweise die Grundlagen der Lichtausbreitung formalisiert hat. So wird es möglich, architektonische Abstraktionen - etwa ein geometrisches Modell - mit physikalischen Abstraktionen zu kombinieren und als Resultat eine photorealistische Simulation zu erzeugen. Die Simulation verschiedener Lichtquellen, Schatten und Materialien wird dadurch nach unter-

Drahtmodell (Wireframe) des
Architectural Space Laboratory (ASL).
Wenz, 1992



ASL einfach schattiert - Flat Shading.
Van der Mark, 1993



schiedlichen Verfahren möglich. Wie die traditionelle Präsentation ist auch die Präsentation mit den neuen Mitteln eine Kunst, die wenige meistern. Photorealistische Darstellungen oder Renderings sind die vorläufig letzte Stufe in der architektonischen Präsentation und haben meist den Zweck, eine wirklichkeitsnahe Darstellung für die nicht an architektonische Abstraktionen gewöhnten Auftraggeber oder Entscheidungsträger zu produzieren. Die Kunst der photorealistischen Darstellung ist sehr spezialisiert und der architektonischen Abstraktion in der Präsentation entgegengesetzt. Architektonische Abstraktionen erlauben schnellere und präzisere Manipulationen, Renderings dagegen zeigen ein anscheinend endgültiges Stadium des Entwurfs mit geringem Abstraktionsgrad. Sie sind deshalb wesentlich schwieriger und langsamer zu manipulieren. Das Verlangen der Designer geht in Richtung einer nahtlosen Integration von Skizzieren, Modellieren und Rendering in einem durchgehenden Computerprogramm. Leider befriedigt keines der jetzt existierenden kommerziellen Programme diese Bedürfnisse vollkommen, was wiederum ihre Akzeptanz für Entwurfsaufgaben beeinträchtigt. Photorealistische Darstellungen werden langsam



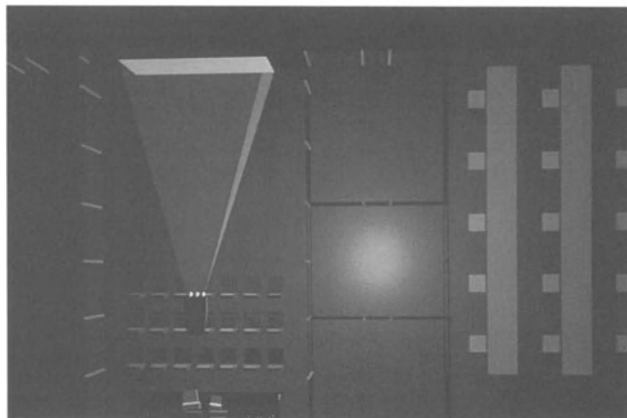
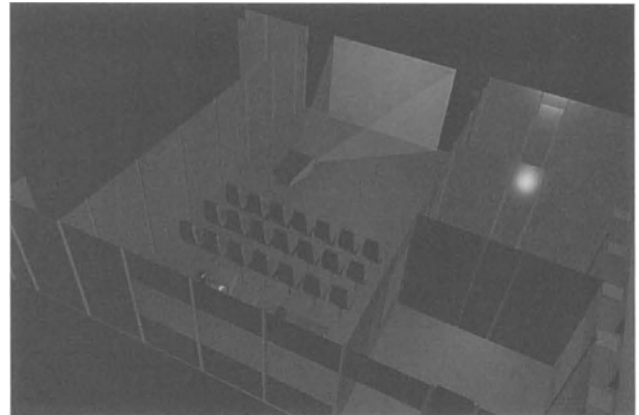
zur Konkurrenz für dreidimensionale, physische Modelle, die einen bedeutenden Kostenfaktor bei Wettbewerben und Entwürfen darstellen. Der Vorteil von Renderings ist normalerweise eine kürzere Erstellungszeit und ein höherer Grad von Material-Realismus. Nachteile sind der Verlust der dreidimensionalen und haptischen Qualitäten von gebauten Modellen.

Zusätzlich zu den statischen Präsentationen werden mit dem Computer verstärkt Animationen möglich. Der Einsatz von computergenerierten Videos für bestehende Architektur in Form von Modelldurchgängen und Modellüberfliegungen hat eine lange

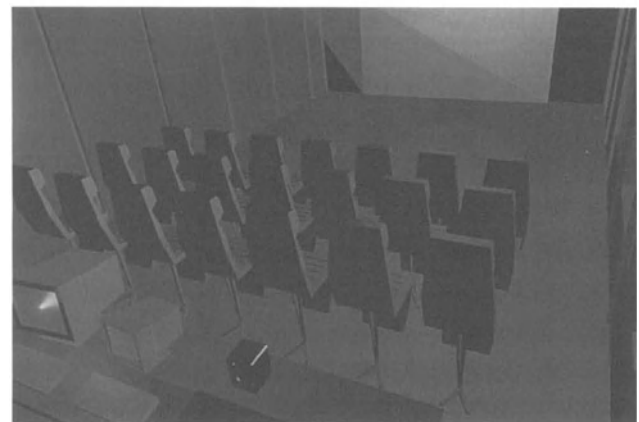
ASL photorealistische Darstellung.
 Van der Mark, 1993



ASL Einzelbilder aus einem Video.
 Wenz und van der Mark, 1993



Geschichte. Am Anfang standen Videos computergenerierter Drahtmodelle. Es folgten Animationen mit einfachen Flächenmodellen. Heute werden Bewegungssequenzen in einzelnen Rahmen oder Frames von photorealistischen Darstellungen von Computern gerechnet und automatisch auf Video aufgezeichnet. Videos von qualitativ hochstehenden Renderings sind noch kein Präsentationsmedium für das durchschnittliche Büro, sondern eher für Dienstleistungsbetriebe der Graphikbranche. Die Kombination realistischer Szenerie und simulierter Architektur ist zu einer weithin akzeptierten Darstellungsform geworden, um die Qualität eines vorgeschlagenen Gebäudes im ge-

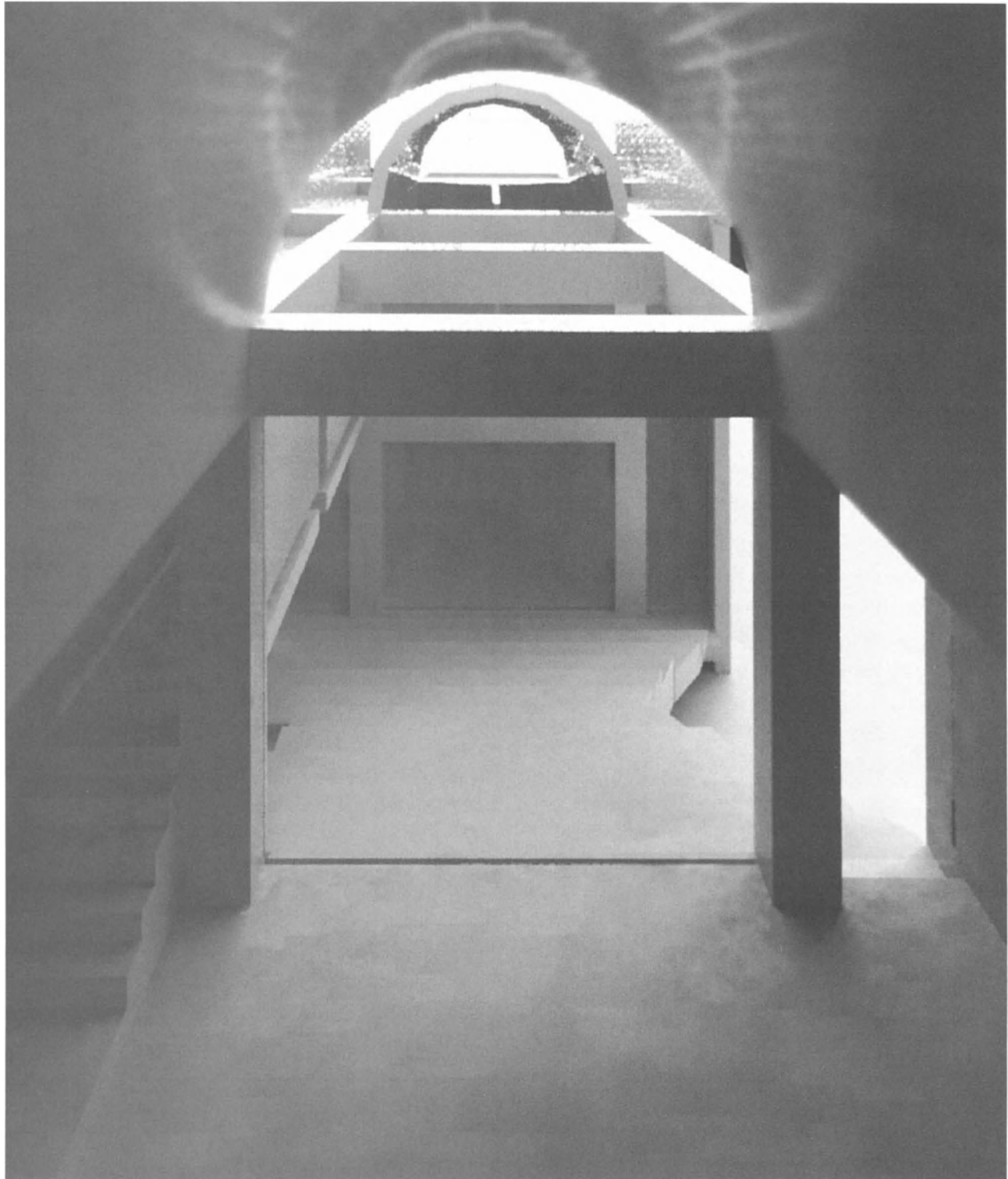


benen Kontext zu zeigen und Faktoren wie Maßstab und Benutzerverhalten zu demonstrieren.

Bei der Modellierung neuer Räume und Gebäude wäre es notwendig, diese Art von Visualisierung bereits in der Entwurfsphase und in Echtzeit zur Verfügung zu haben. Voraussetzung dazu ist eine Steigerung der Geschwindigkeit von Workstations in den Gigaflop-Bereich. Zu Beginn der neunziger Jahre leisten Workstations lediglich einige Megaflops. Detail einer Diplomarbeit an der ETH Zürich. Mischler, 1993



Innenraum des 'Matsumoto House' von Tadao Ando, dargestellt mit dem Programm Radiance. Der Weg vom CAD-Modell zum fertigen Rendering ist immer gleich und auf den vorhergehenden Seiten angedeutet:
(1) Erzeugung eines dreidimensionalen Modells, (2) Export der Daten in ein Visualisierungs- oder Rendering-Programm, (3) Definition der Materialien und Lichtquellen (immer häufiger bereits im CAD-Programm möglich), (4) Berechnung der photorealistischen Darstellung. Modell von der Mark und Vezin, Rendering von Lucius, 1993



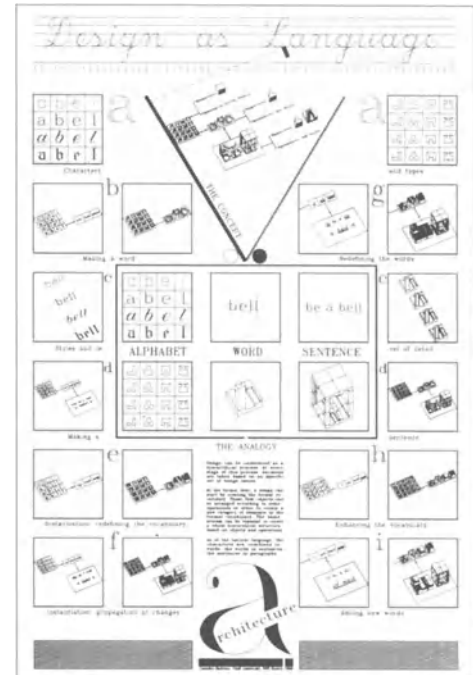
Principia: Entwurf als Sprache

Die Beziehung zwischen Architektur und natürlicher Sprache, bekannt unter dem Begriff 'Design as Language', ist seit Jahren Gegenstand der Forschung. Besonders Anfang der achtziger Jahre, angeregt durch neue Forschung in der Psychologie und durch die Computerimplementierung generativer Algorithmen, gewann die Idee von Architektur als Sprache an Boden. Mitchell faßte diese Gedanken zusammen [Mitchell 1990a].

Er versuchte zunächst, Gebäude mit Wörtern zu beschreiben und demonstrierte danach, wie solche Beschreibungen mit Prädikatenlogik erster Stufe (First-Order Predicate Calculus) formalisierbar sind. Das führte ihn zu einem Modell, das aus dem Betrachter, einem Entwurfsraum und einem Sprachraum besteht. In den Entwurfsraum setzte Mitchell graphische Symbole, die nach grammatischen Regeln manipuliert werden dürfen. Er zeigte, wie Regionen dieses Entwurfsraumes durch Formengrammatiken definierbar sind. Er verband den Sprachraum und den Entwurfsraum dadurch, daß er zeigte, wie Sprache als Entwurf

im Entwurfsraum interpretierbar waren. Entwurf wurde dadurch im Entwurfsraum berechenbar mit der erklärten Absicht, die in der Sprache ausgedrückten formalen und funktionalen Bedingungen zu erfüllen. Nehmen wir das Beispiel der Sprache auf, so drängt sich der Gedanke auf, daß es analog zu Textverarbeitungssystemen auch Entwurfsverarbeitungssysteme geben müßte. Doch würden diese die Qualität der Architektur verbessern? Es ist unbestreitbar, daß heute auch die besten Schriftstellerinnen und Schriftsteller Textverarbeitungssysteme benutzen, sie sind also zumindest kein Hinderungsgrund für ihre Arbeit. Doch der Hauptunterschied ist, daß Textverarbeitungssysteme lediglich ein beschränktes Vokabular - Buchstaben und eine begrenzte Anzahl von Zeichen - zu verarbeiten haben, während ein Designvokabular, falls nicht willkürlich beschränkt, unendlich viele Elemente enthält.

In der Lehre ist es anfangs sinnvoll, die syntaktischen und semantischen Analogien von Entwurf und Sprache voneinander zu trennen. Seit 1989 haben wir des-



Design as Language,
 Plakat für das Sommersemester 1991.
 Madrazo, 1991

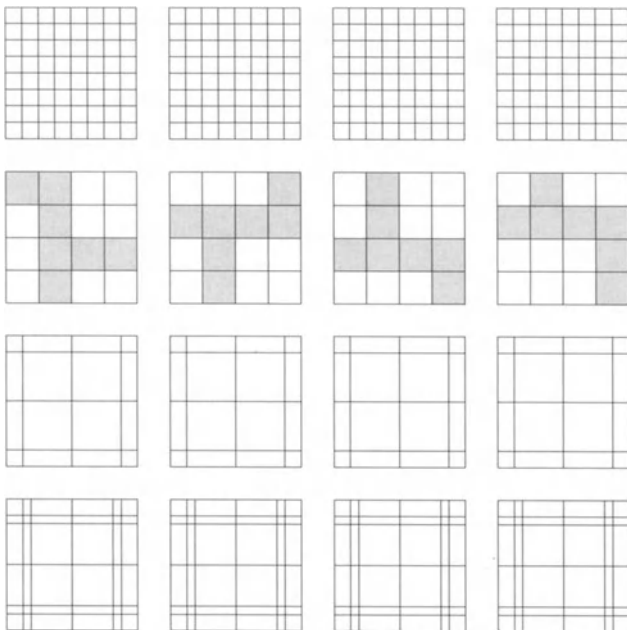
halb verschiedene Programme implementiert, um die Beziehung zwischen Entwurf und Sprache als Ordnungselement zu nutzen [Madrazo 1992a]. Daraus entstand 1990 der erste Kurs, der um die Analogien zwischen Sprache, Geometrie und deren Verarbeitung im Computer aufgebaut ist. Voraussetzung war die Schaffung von Programmen, welche gewisse Entwurfsmethoden direkt unterstützen. Das erste Ziel des Kurses ist das Lehren von Entwurfsprinzipien, die zu Beginn von spezifischen architektonischen Fragestellungen losgelöst sind. Dieses Vorgehen hat sich in unserer Erfahrung bei der Verwendung von Computern im Architekturunterricht obwohl die meisten kommerziellen Programme nicht in der Lage sind, architektonische Gesichtspunkte direkt und über syntaktische Operationen hinausgehend zu behandeln, unterstützen Computerinstrumente geometrische Operationen ausgezeichnet.

Der Kurs basiert auf der objektorientierten Modellierweise (siehe Instrument: Objektorientiertes Modellieren I-IV) und nutzt dementsprechend die Organisati-

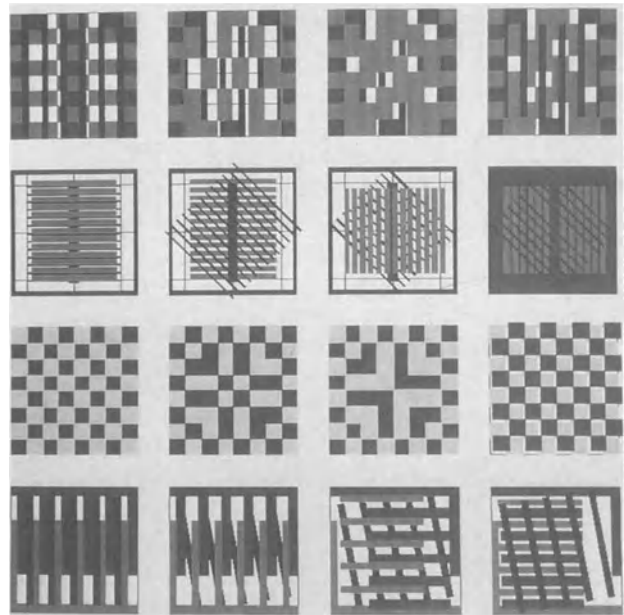
on und Manipulation von Objekten als Typen und Variationen. Zu Beginn repräsentiert ein Vokabular ausgezogener Profile die Typen, die durch interaktive Editierinstrumente einfach manipulierbar sind. Mit Variationen dieser Typen bearbeiten die Studentinnen und Studenten die ersten Übungen. Spezielle Funktionen erlauben die Änderung der Objektattribute, wodurch formale Änderungen sehr schnell ausführbar sind. Als zweiten Schritt führen wir die Substitution ein, mit deren Hilfe aus verschiedenen Typen bestehende Objekte verändert werden. Als dritten Schritt lernen die Studentinnen und Studenten die Möglichkeiten kennen, welche die Erhöhung oder Reduzierung des Detaillierungsgrades bieten. In einer vierten Übung beginnen sie mit der hierarchischen Kombination verschiedener Typen in zusammengesetzte Objekte, die wir Compound Objects nennen. Den Abschluß des Semesters bildet die Anwendung dieser Prinzipien auf einen konkreten Entwurf, worin der Schritt zur Architektur geschieht. Ein Beispiel ist der Entwurf einer Bibliothek.

Schritt 1: Vokabular und erste Kompositionen auf der Ebene

Vorgabe: 16 Quadrate mit Rastern.
Madraza, 1991



Komposition mit vier Elementen.
Racine, 1993

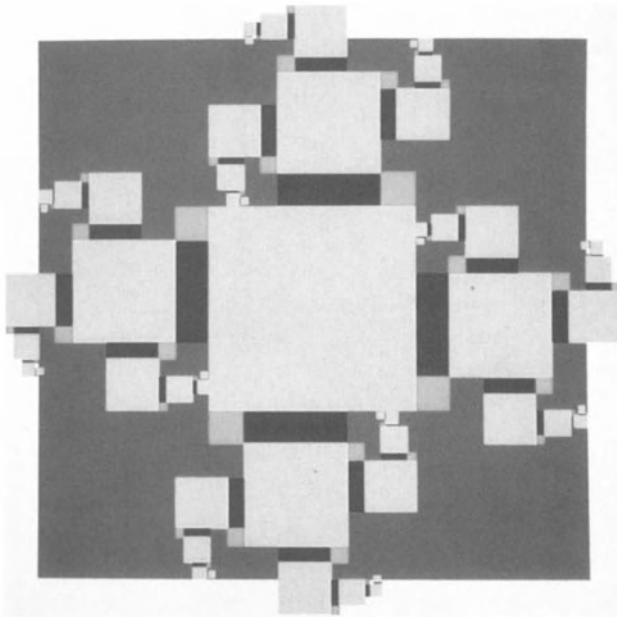


Im ersten Schritt erhalten die Studierenden eine Sammlung einfacher Objekte, die zusammen das Vokabular bilden. Die Objekte unterscheiden sich nach ihrer Farbe. Jedes einzelne Objekt bezeichnen wir als Typ (siehe Instrument: Objektorientiertes Modellieren I - Typen und Variationen). Als weitere Vorgabe erscheint ein Raster von 16 Quadraten, auf denen vier Reihen von Kompositionen zu entwickeln sind. Die ursprüngliche Komposition befindet sich jeweils in der ersten Spalte, in den weiteren Spalten sind Variationen der vorhergehenden Komposition zu entwerfen. Die Komposition nutzt lediglich das in Form von Typen abgespeicherte Vokabular. Allerdings können die Typen in Länge und Breite variiert und

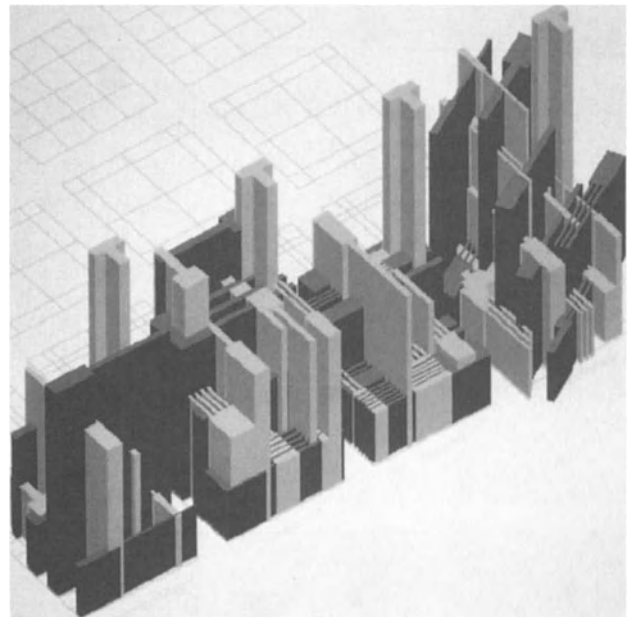
gedreht werden. So werden die Proportions- und Rotations-Parameter erkannt. Einmal plaziert, kommen die Spiegelung, das Kopieren und Verschieben hinzu. Die Studentinnen und Studenten lernen, mit einfachen parametrisierten Typen eine Vielzahl von Variationen herzustellen, die zudem als Gesamtkomposition eine gestalterische Aussage macht.

Die Fähigkeit, Elemente zu organisieren und mit ihnen in einer klaren und bewußten Art umzugehen, ist ein Charakteristikum des Entwurfsprozesses. Wir versuchen so zu demonstrieren, daß selbst mit einem kleinen Vokabular systematische und zugleich kreative Kompositionen möglich sind.

Komposition mit drei Elementen.
Colazzo, 1993



Komposition dreidimensionaler Elemente auf
der Ebene. Gyarfas, 1992



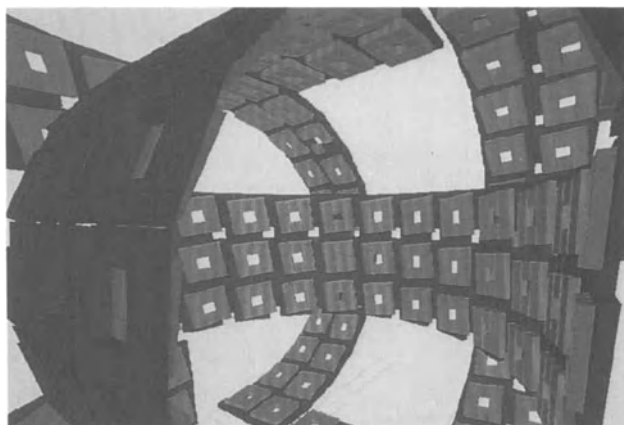
In einer Variante der ersten Übung erzeugen die Studentinnen und Studenten die Formen selbst, indem sie einen von uns programmierten Formengenerator benutzen. Dieser basiert auf einem 3x3-Raster. Durch Anklicken der Felder wird die Form bestimmt, wodurch U-, L-, T- und viele andere Objekte entstehen können. Das Programm wandelt nach abgeschlossener Selektion diese zunächst zweidimensionalen Elemente in dreidimensionale Körper um und addiert sie zu einer Bibliothek der Typen. Auf diese Weise entwickelt sich das Vokabular verschiedener Typen. Diese stehen danach wie in der zuvor beschriebenen Übung als Elemente einer Komposition zur Verfügung. Die Elemente werden auf einer Ebene plazierte und können dort nach ihrer

Länge, Breite, Höhe und Einsetzwinkel variiert werden. Die Resultate sind wesentlich reicher an Information. Allerdings ist die Beziehung der Variationen zum ursprünglichen Typ nur noch dann erkennbar, wenn die Objekte frei stehen und in ihren Proportionen nicht wesentlich verändert wurden. Bei Kombinationen von Typenvariationen entstehen neuartig wirkende Elemente. Die bewußte Verfolgung dieses Phänomens geschieht in der späteren Übung mit Compound Objects.

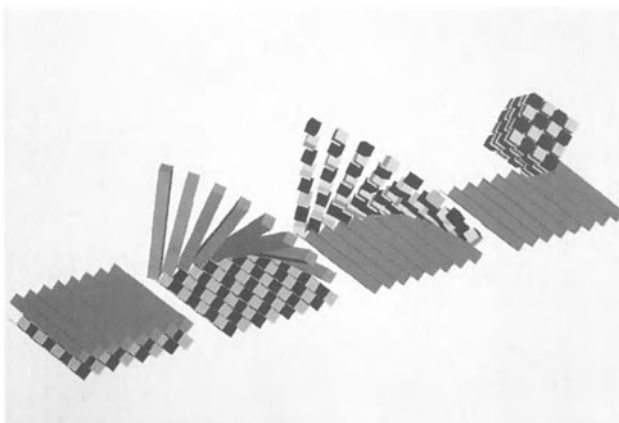
Diese Übung haben wir in verschiedenen Semestern verfeinert und es zeigt sich, daß die Ergebnisse der vorhergehenden Jahre die neuen Arbeiten in positiver Weise beeinflussen. So ist das, was zunächst exotisch erschien, zu einer beliebten Aufgabe geworden.

Schritt 2: Konstruktion im Raum und Substitution

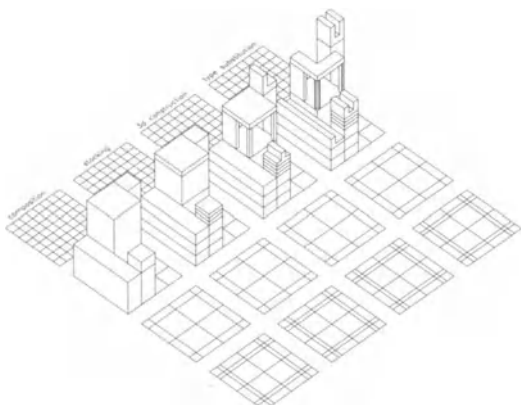
Konstruktion im Raum mit zwei Elementen.
Grohe, 1992



Konstruktion im Raum mit vier Elementen.
Keel, 1993

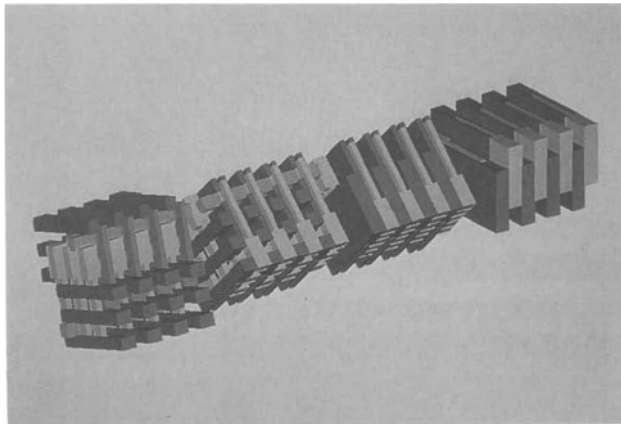


Vorgabe: Einfache Substitutions-
Beispiele. Madrazo, 1991

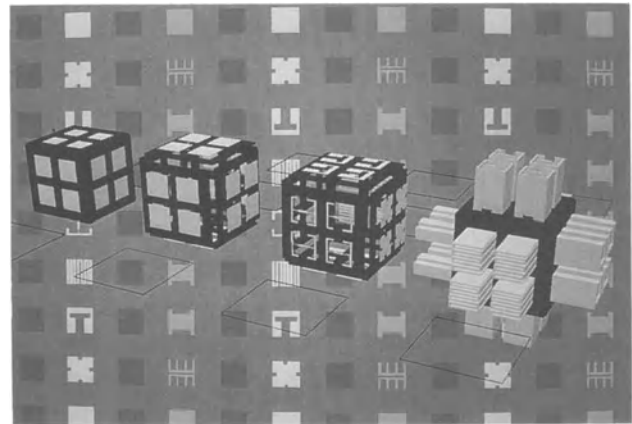


Die Konstruktion im dreidimensionalen Raum, aus dem traditionellen Entwerfen bekannt, ergänzen wir durch die nur mit dem Computer mögliche Fähigkeit der Substitution (siehe Instrument: objektorientiertes Modellieren II - Substitution). Bei der Verwendung traditioneller Entwurfsmethoden ist die Darstellung auf dem Medium Papier immer zweidimensional, das entwerferische Denken dagegen dreidimensional. In einem räumlichen Modellersystem ist das Objekt selbst dreidimensional und erlaubt so beliebige zweidimensionale Abbildungen auf dem Bildschirm. Das Modellieren gewinnt in diesem Schritt mehr Freiheitsgrade und verlangt daher eine höhere Konzentration. Neue Operatoren gelangen zum Einsatz. Typenvariationen lassen sich nun im Raum und um jede Achse bewegen, drehen, skalieren und spiegeln. Das Verlassen der Ebene, des sicheren Bodens, bereitet vielen Studentinnen und Studenten zu Beginn gestalterische

Konstruktion im Raum und schrittweise Substitution von Elementen. Hembus, 1993



Konstruktion im Raum und Substitution vor dem Hintergrund der verwendeten Typen. Cavero, 1992

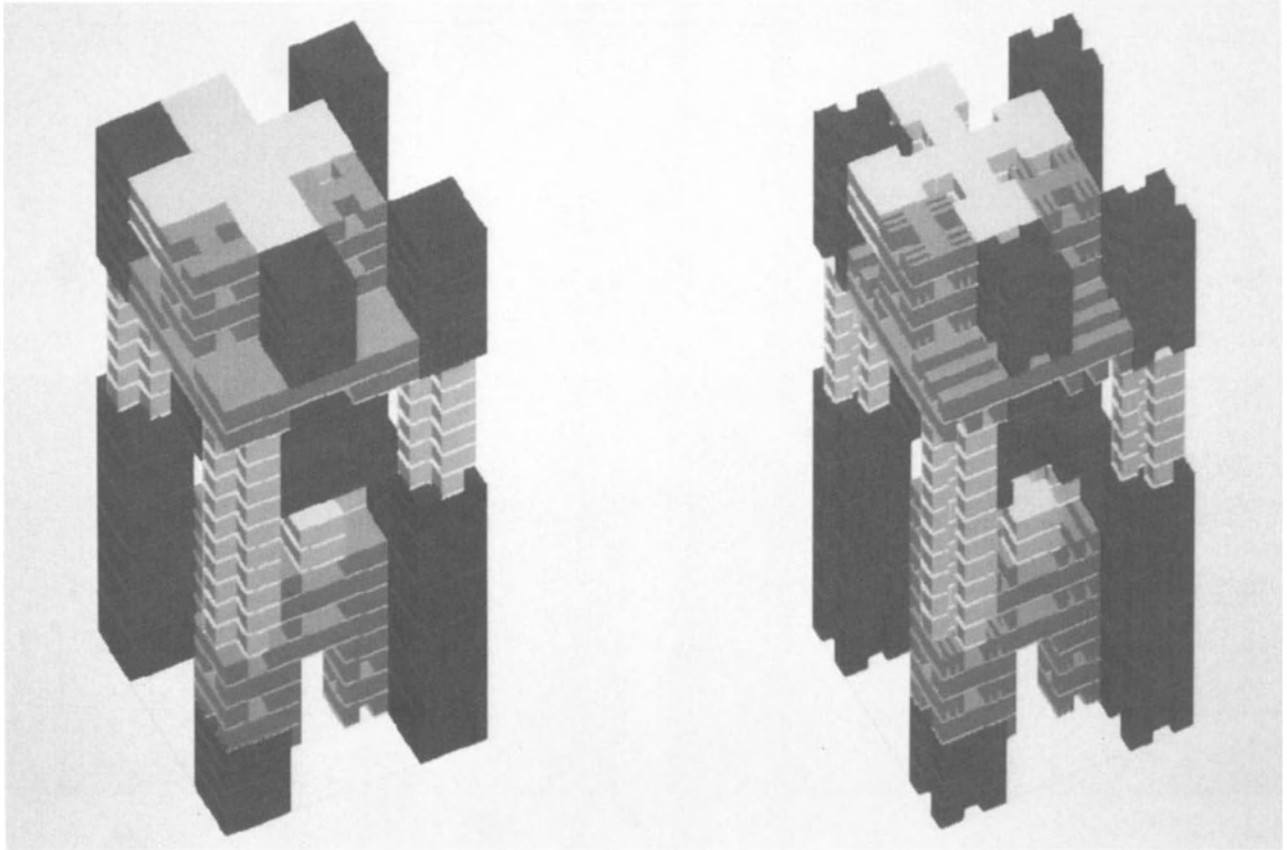


Mühe, sie lernen jedoch schnell die neuen Freiheiten zu schätzen. In der ersten Übung verwenden die Studentinnen und Studenten Orthogonalprojektionen und Axonometrien die das Objekt auf eine abstrakte Art zeigen. Der Augpunkt liegt im Unendlichen; es besteht so eine große Distanz zwischen Entwerfer und Objekt. Bei Verwendung der Perspektive befinden wir uns dagegen in einem festgelegten Abstand vom Objekt, wodurch sich eine Beziehung zwischen Entwerfer und Objekt aufbaut.

In dieser Übung führen wir das Prinzip der Substitution ein. In der natürlichen Sprache bestimmt das Verhältnis zwischen den Satzteilen die Struktur einer Sprache. Die Worte eines Satzes lassen sich durch andere Worte so ersetzen, daß dadurch die syntaktische - seltener die semantische - Richtigkeit des Satzes bestehen bleiben kann. Damit erweist sich die Struktur eines Satzes als wichtige Komponente einer Sprache (siehe

Instrument: Generieren I - Formengrammatiken). Dieses linguistische Modell wenden wir hier analog an. In der klassischen Architektur gibt es eine Trennung zwischen den sichtbaren Elementen (entsprechend den Wörtern einer Sprache) und der nicht sichtbaren Struktur (entsprechend den syntaktischen Regeln). Die sichtbaren Elemente einer Säule, - Basis, Schaft und Kapitell- ordnen sie einem bestimmten Stil zu. Die nicht sichtbare Struktur der Säule ist durch die Beziehung der drei Teile zueinander gekennzeichnet. Diese Unterscheidung zwischen Form und Struktur ist fundamental wichtig für den Einsatz des Computers im Entwurf und insbesondere für die Verwendung der Substitution. Nur nachdem Studierende die Struktur eines Objektes klar erkennen, ist es sinnvoll, einzelne seiner Teile durch andere zu ersetzen. Wird diese Beziehung nicht erkannt, verkommt die Substitution zu einem wahllosen Austauschen von Typenvariationen.

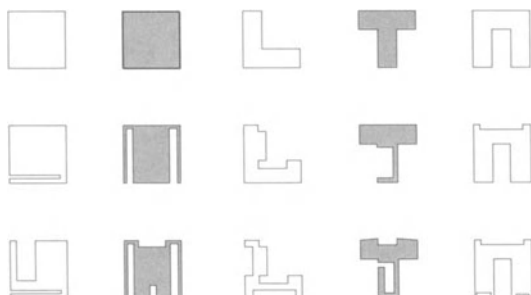
Schritt 3: Multiple Repräsentation und Detaillierungsgrad



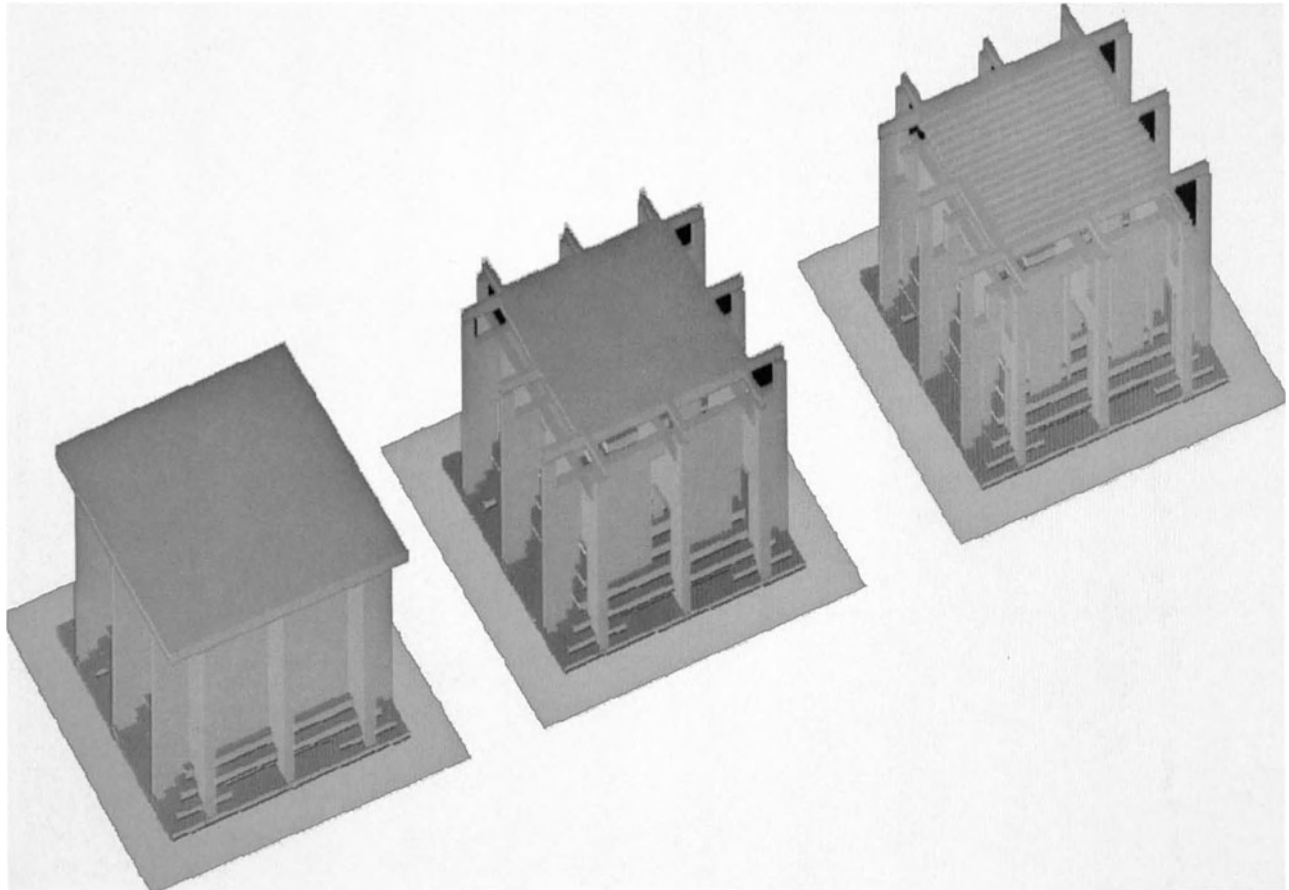
Erhöhung des Detaillierungsgrades von links nach rechts. Gyrfas, 1992

Während des Entwurfens ist es wichtig, daß für Objekte verschiedene Abstraktionsgrade zur Verfügung stehen (siehe Methode: Abstraktion und Modellbildung). Wir wollen in der Lage sein, ein Objekt nur in seinen räumlichen Umrissen zu sehen, in einer vorläufigen Darstellung oder in verschiedenen Graden der Detaillierung. Dabei gehen wir davon aus, daß grundsätzliche Entscheidungen über die innere Struktur des Entwurfs bereits wie in Schritt 2 beschrieben gefallen sind. Dieses Vorgehen erlaubt es, Wichtiges von Unwichtigem zu trennen und sich auf ausgewählte Teile des Entwurfs zu konzentrieren, ohne den Gesamtüberblick zu verlieren. Wir erreichen dies mit einem speziell für diesen Zweck entwickelten Instrument (siehe Instrument: Objektorientiertes Modellieren III - Detaillierungsgrade, sowie Instrument: objektorientiertes Modellieren IV - Designfokus - Logical Zoom). Das Äquivalent der Detaillierungsgrade in der natürlichen

Vorgabe: Die Typen und Detaillierungsgrade. Madrazo und Matter, 1990



Schrittweise Erhöhung des Detaillierungsgrades zur Darstellung von Alternativen. Grohe, 1992

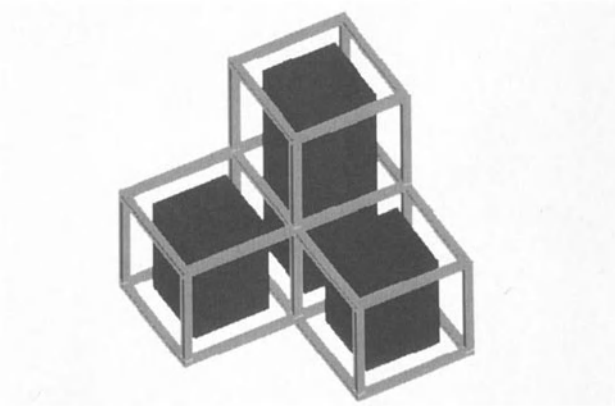


Sprache sind die verschiedenen Sichten, unter denen man ein Dokument betrachten kann. Textverarbeitungsprogramme zum Beispiel bieten die Funktion der Gliederung an. Damit ist es möglich, in einem Text verschiedene Hierarchien einzuführen und ein Dokument in Kapitel, Abschnitte und Unterabschnitte zu teilen, die vollständig oder unabhängig voneinander verschoben oder in einem höheren oder niedrigeren Detaillierungsgrad betrachtet werden können. So kann man ein Buch nur nach Kapitelüberschriften, oder Abschnittüberschriften, oder ausgewählten Texten lesen. Das Herstellen der Detaillierungsgrade in unserem System verläuft denkbar einfach. Die Typen bilden die höchste und meist auch abstrakteste Darstellungsart. Sie verhalten sich wie Kapitel in einem Inhaltsverzeichnis. Jeder beliebige Typ kann einzeln bearbeitet und auf einen höheren Detaillierungsgrad gebracht werden. Dazu erscheint der Typ in einem separaten

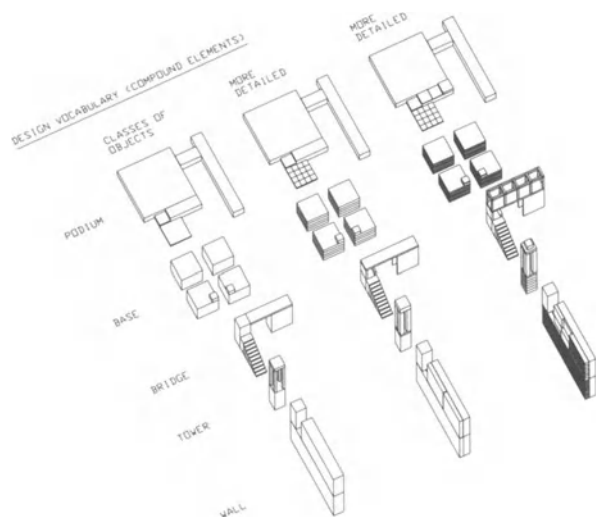
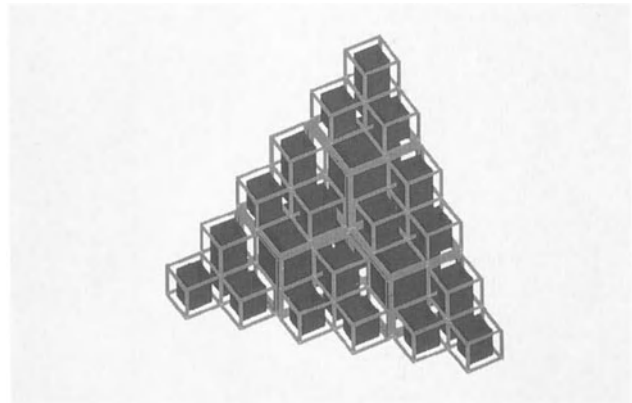
Fenster. Der Entwerfer wendet die bekannten geometrischen Manipulationen auf das Objekt an, ändert also beispielsweise das Profil leicht ab oder fügt zusätzliche Informationen ein. Es ist ratsam, den ursprünglichen Typ nicht bis zur Unkenntlichkeit zu verfälschen, da sonst die Beziehung zwischen Typ und Detaillierungsgraden optisch nicht mehr gewährleistet ist. Sind alle gewünschten Verfeinerungen angebracht, speichert das Programm den Detaillierungsgrad als 'Kind' des ursprünglichen Typs ab, das wichtige Eigenschaften wie den geometrischen Ursprung, Orientierung, Material und anderes 'erbt'. Nur so ist es später möglich, den Detaillierungsgrad in einem Modell schnell und elegant zu ändern und zu garantieren, daß die Gesamtgeometrie nicht zerstört wird. Im Unterschied zur Substitution, die ein Austauschen von Objekten innerhalb einer Struktur erlaubt, zielt die Änderung des Detaillierungsgrades also lediglich auf eine Verfeinerung.

Schritt 4: Hierarchien und Compound Objects

Zusammenfassung von Elementen als Compound Object. Eglin, 1993



Kombination des Compound Objects in eine Hierarchie. Eglin, 1993

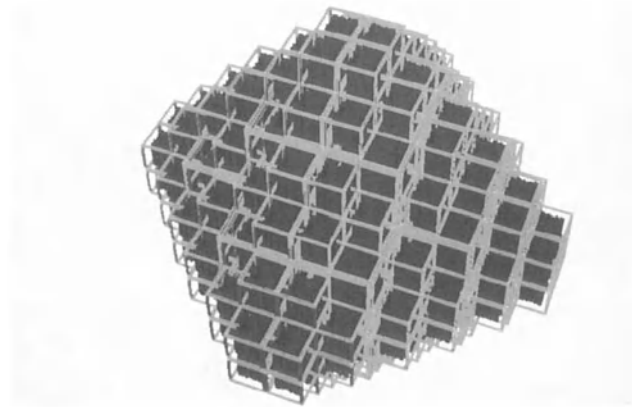


Vorgabe: Beispiele für Compound Objects. Madrazo, 1991

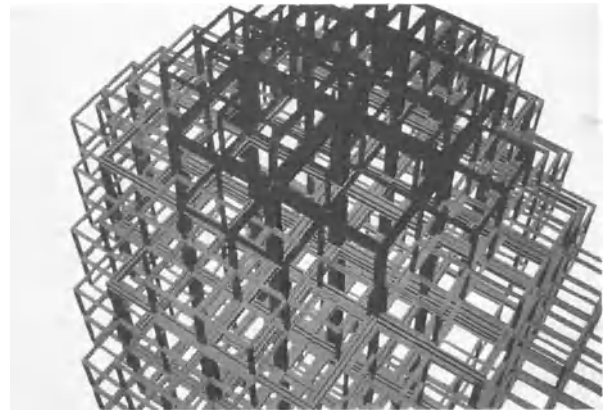
Der Sinn dieses Schrittes besteht darin, zusammengehörige Elemente innerhalb eines Objektes zu erkennen und diese so zu gruppieren und zu verbinden, daß sie als Einheit behandelt werden können. Damit lernen die Studentinnen und Studenten, Hierarchien zu erkennen, beziehungsweise zu etablieren. Die so definierten hierarchischen Objekte verhalten sich danach wie die zuvor beschriebenen Typen. Sie können ebenso gespeichert und abgerufen werden. Allerdings kann ein zusammengesetztes Objekt Elemente auf verschiedenen Detaillierungsgraden enthalten, was bei der späteren Weiterverarbeitung der Variationen besondere Sorgfalt verlangt.

Die Analogie zur Sprache wird in diesem Gebiet gewagter. Man könnte noch die Zusammensetzung von Wörtern zu Sätzen oder von Sätzen zu Abschnitten heranziehen. Nützlicher ist dagegen wieder das Beispiel der klassischen Säule. Die innere Zusammen-

Bildung einer zweiten Hierarchie mit Compound Objects. Eglin, 1993



Detailansicht der zweiten Hierarchie nach Ausblendung der Innenvolumen. Eglin, 1993



gehörigkeit der Teile oder Typen - Basis, Schaft und Kapitell - dokumentieren wir durch die Definition der gesamten Säule als hierarchisches Objekt oder als Compound Object. Als solches wird es gespeichert, und als solches ist es später wieder abrufbar.

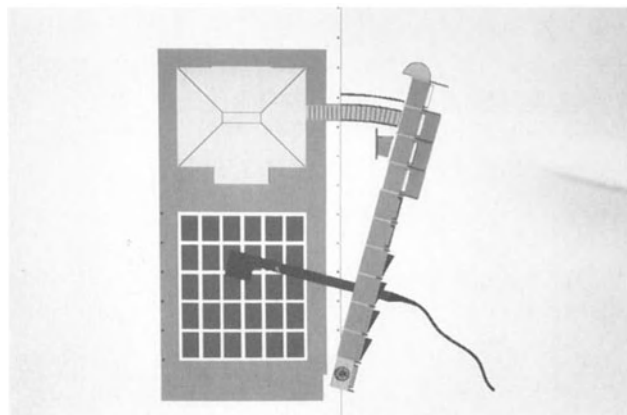
In einer typischen Übung gestalten die Studentinnen und Studenten zunächst eine räumliche Komposition mit den bereits bekannten Mitteln. Danach nehmen sie innerhalb der Komposition die entsprechenden Gruppierungen vor und definieren so die zusammengesetzten Objekte. In einem weiteren Schritt erzeugen sie zwei Variationen aus den neu definierten Compound Objects, die deutliche Unterschiede zur ursprünglichen, nicht hierarchisch gegliederten Komposition zeigen.

Der Unterschied zwischen einfachen Typen und Compound Objects ist eher praktischer als theoretischer Art, denn mit dem Ordnungsprinzip der Compound

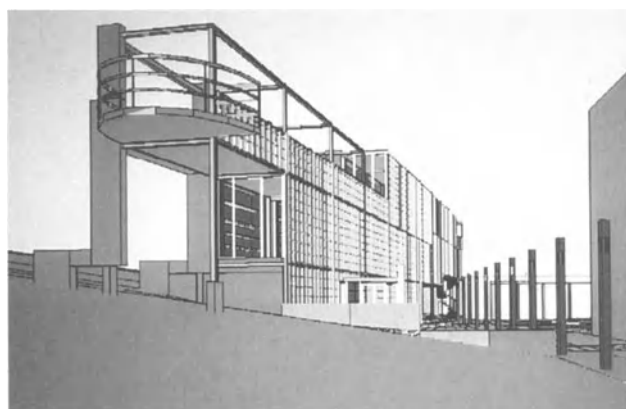
Objects läßt sich eine weitere Ebene in die hierarchische Organisation der Objekte einführen. Spätestens bei dieser Übung wird auch der Bezug zur Architektur deutlich. Zusammengehörige Bauteile lassen sich so gut modellieren und vor allem auch als Einheit in der Datenbank abspeichern und später wieder abrufen. Sie eignen sich hervorragend zum Studium von Alternativen in einem Modell. Wir können beobachten, daß mit dieser zweiten Ebene von Organisation die Grenze des einfach Nachvollziehbaren bei den Studierenden erreicht ist. Da sie in der Lage sind, beliebige Elemente aus dem Modell zu übernehmen und als Gruppe zu definieren, nehmen die Compound Objects schnell erheblichen Umfang an. Das Organisationsprinzip der Compound Objects ist in kommerziellen CAD-Programmen in unterschiedlicher Tiefe zu finden, am ehesten bei den auf objektorientierten Programmier-techniken basierenden Anwendungen.

Principia: Objektorientiertes Entwerfen

Lageplan mit bestehender Villa.
Zimmermann, 1992

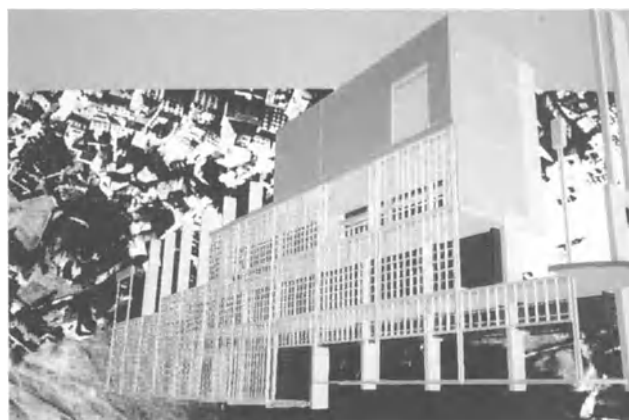


Perspektive von Osten (oben) und von Westen (unten). Das Modell entstand mit den Instrumenten 'Typen und Variationen' und Prototypen. Zimmermann, 1992



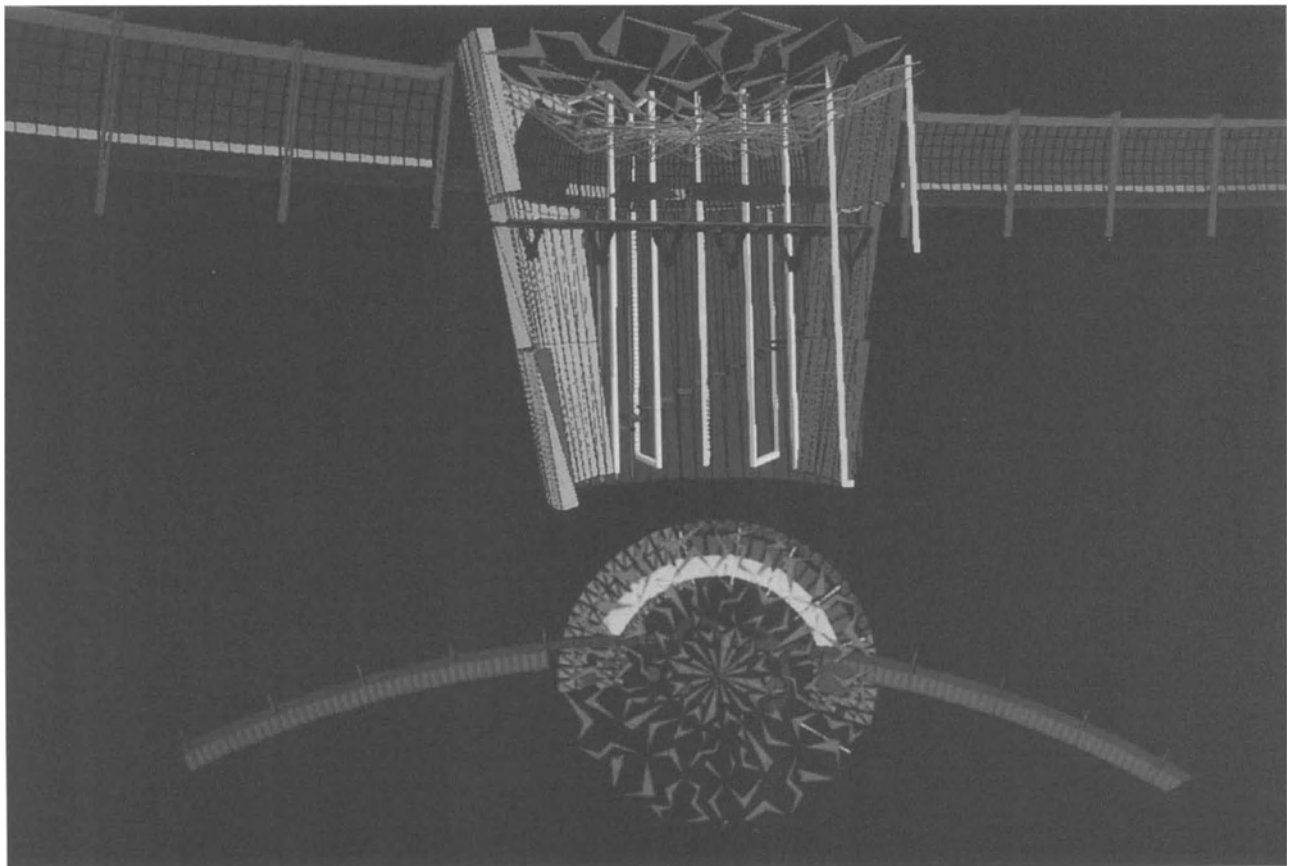
Sämtliche im vorangegangenen Abschnitt angesprochenen Übungen bleiben auf das Arbeiten mit Objekten beschränkt. Verbindungen zum Entwurf werden angesprochen, die Übungen sind aber nicht primär zur Erzeugung architektonischer Objekte gedacht. Denn Entwurf als Sprache ist lediglich eine Sichtweise und ein gedanklicher Rahmen, den das objektorientierte Modellieren unterstützt. Die Kenntnis des beispielhaft beschriebenen Vorgehens dient vielmehr bei der Anwendung im Entwurf als wichtige methodische Grundlage. Die syntaktischen Aspekte des Entwurfs stehen dabei im Vordergrund. Doch auch auf der Ebene der Instrumente selbst - also der Typen und Variationen, der Substitution, der Detaillierungsgrade und der Compound Objects - sind interessante Entwurfsanstöße möglich.

Das folgende Projekt des Entwurfs einer Bibliothek dient als Beispiel. Gegeben waren das Baugrundstück mit architektonischem Kontext sowie das Bauprogramm. In der ersten Phase des Semesters lernten die Studentinnen und Studenten die Methode der Prototypen, das Instrument der Parametrisierung und das objektorientierte Modellieren kennen.



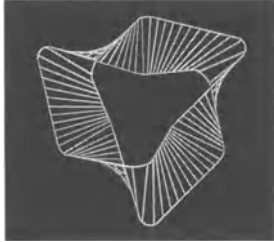
Die zweite Phase war dem Entwurf der Bibliothek gewidmet. Der erste Schritt war die Erzeugung von Massenmodellen mit einem von Sharon Refvem entwickelten Massenmodell-Programm. Das Programm kannte die geforderten Räume, deren minimale und maximale Kubatur, Ausdehnung und Höhe und verfügte über die Fähigkeit, verschiedene Alternativen zu evaluieren, zu speichern und wieder abzuspielen. Die Platzierung des Massenmodells geschah direkt im dreidimensionalen Kontextmodell. Die so geschaffenen Alternativen gaben darüber Auskunft, in welchem Ausmaß sie die

Vorschlag für eine unterirdische Bibliothek in Einsiedeln im Garten der Villa. Freie Verwendung der Instrumente. Bal, 1992



Programmanforderungen erfüllten. Auch ihr Verhältnis zu der auf dem Grundstück stehenden Villa kam klar zum Ausdruck. Innerhalb des so entwickelten Massenmodells konnte der Entwurf danach vertieft werden. Mit Typen und Variationen platzierten die Studentinnen und Studenten zunächst wichtige Räume und Konstruktionen. Mit der Substitution innerhalb der so entstandenen Struktur studierten sie Alternativen. Danach benutzten sie Detaillierungsgrade, um ausgewählte Architekturelemente weiter zu entwickeln. Schließlich gestattete die Benutzung von zusammengesetz-

ten Objekten die hierarchische und logische Gliederung des Gebäudes. Die räumliche Überprüfung des Resultats erfolgte durch die ständige Verwendung eines Visualisierungsprogramms. Deutlich wurde auch, daß die Studierenden die von uns programmierten Entwurfshilfen anders als ursprünglich beabsichtigt verwendeten. Erstaunlich war der Grad der Unterschiede in den Endergebnissen, denn trotz der Verwendung derselben Instrumente waren die Projekte in Form und Inhalt weit voneinander entfernt und zeugten vom innovativen Einsatz der neuen Instrumente.



Entwurf einer Bibliothek in Einsiedeln. Prinzip der Möbius-Kurve (oben). Inneres modelliert unter Verwendung von Typen und Variationen sowie parametrisierten Büchergestellen. Modell und Rendering Sladoljev, 1992



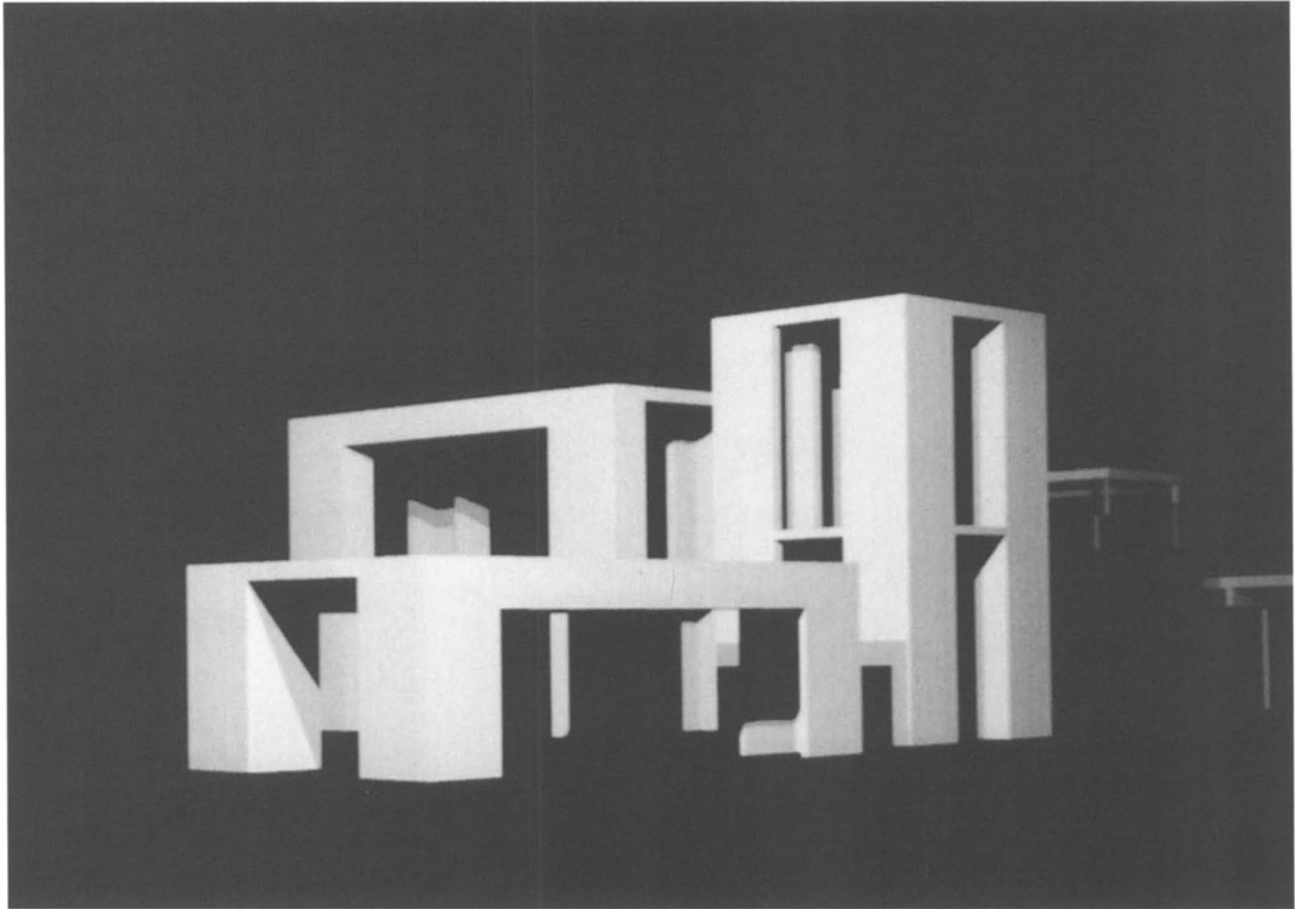


Entwurf einer Bibliothek in Einsiedeln. Außenansicht von Nordwesten (oben). Blick in das Obergeschoß. Modell und Rendering Stadoljev, 1992



Principia: Modellieren mit Prototypen

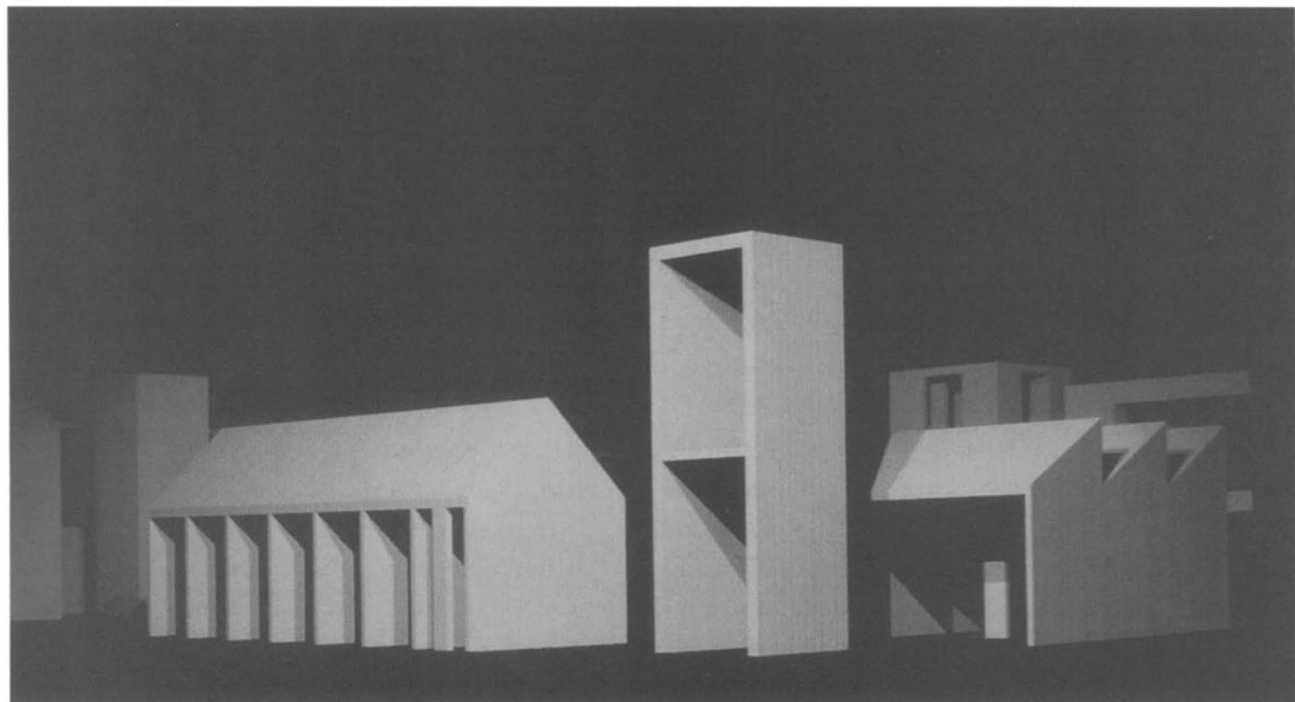
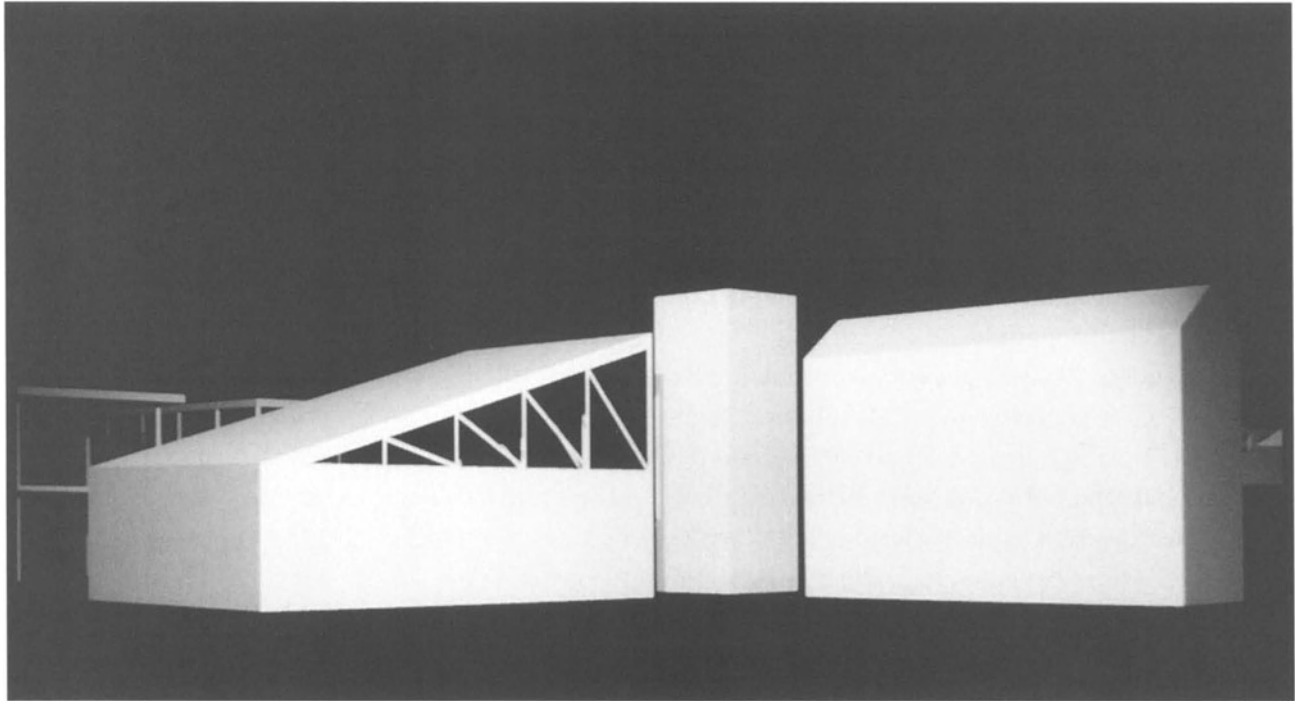
Objekte nach dem Lars-Lerup-Prototyp. Unterzüge werden bei Überschreiten der zulässigen Spannweite der Betondecken eingefügt.
Regeln: Luebkehan, Programm und Rendering: Engeli, 1992



Modellieren mit Prototypen ist eine Technik, die in Zukunft große Veränderungen in den CAD-Systemen verlangen wird. Im vorliegenden Beispiel haben wir uns typische Konstruktionen von Rossi, Sterling, Lerup und Le Corbusier vorgenommen und versucht, mit Hilfe von Chris Luebkehan statische Gemeinsamkeiten zu entdecken und zu formulieren. Ziel war nicht die architektonische Rekonstruktion der Gebäude, sondern die Herstellung eines Computerinstruments, das die Stabilität von automatisch mit Prototypen erzeugten Objekten garantiert, die entfernt an Gebäude der genannten Architekten

erinnern. Nach dieser Definition entstanden vier verschiedene Programme, die von einer graphischen Benutzeroberfläche abrufbar sind. Nach der Wahl eines Prototyps werden bestimmende Parameter, wie Anzahl der Geschosse, Zahl der Achsen, Dachformen und -Neigungen, abgefragt. Danach erscheint ein Set von Rechtecken auf dem Bildschirm, welche die so definierten Constraints darstellen. Innerhalb dieser Rechtecke ist der Entwurf frei und das entsprechende Gebäude wird automatisch erzeugt. Das Programm entscheidet über den Einbau von Unterzügen oder Dachträgern beim Überschreiten gewisser Maße.

Objekte, erzeugt mit Rossi- (oben) und Sterling-Prototypen. Die verschiedenen Dachformen entstehen als Funktion der numerisch und graphisch eingegebene Parameter. Regeln: Luebckeman, Programm und Rendering: Engeli, 1992



Rekonstruktion einer antiken Stadt: Aventicum

Aventicum, das heutige Avenches, liegt am Murtensee in der Westschweiz. Der Name der Stadt ist abgeleitet von der helvetischen Göttin Aventia. Um 8 v. Chr. wurden der Helvetierstaat - Civitas Helvetiorum - und ein neuer administrativer Mittelpunkt, Aventicum, gegründet. Die Wahl des Standortes wurde bestimmt durch die große Durchgangsstraße durchs Schweizer Mittelland, die Nähe des Murtensees und die Qualität des Baugrundes.

Unter Claudius (41-54) änderte Aventicum sein Gesicht. Nach und nach wurden alle Häuser in Stein umgebaut. Der Tod Neros im Jahre 69 stürzte das Reich in politische Wirren, in deren Verlauf die römische Armee die Höfe und Dörfer des Mittellandes plünderte. Tacitus, der einen lebendigen Bericht dieser Ereignisse gab, schilderte den Vormarsch der Truppen gegen Aventicum, das bei dieser Gelegenheit als *Caput Gentis*, Hauptstadt des Volkes, bezeichnet wurde. Unter Vespasian (69-79) wurde Aventicum eine Kolonie. Eine Stadtmauer entstand. Die Wirtschaft erhielt neue Impulse, was sich in vermehrter Bautätigkeit niederschlug. Der Ausbau des Forums mit Tempel und Basilika geschah spätestens in dieser Zeit.

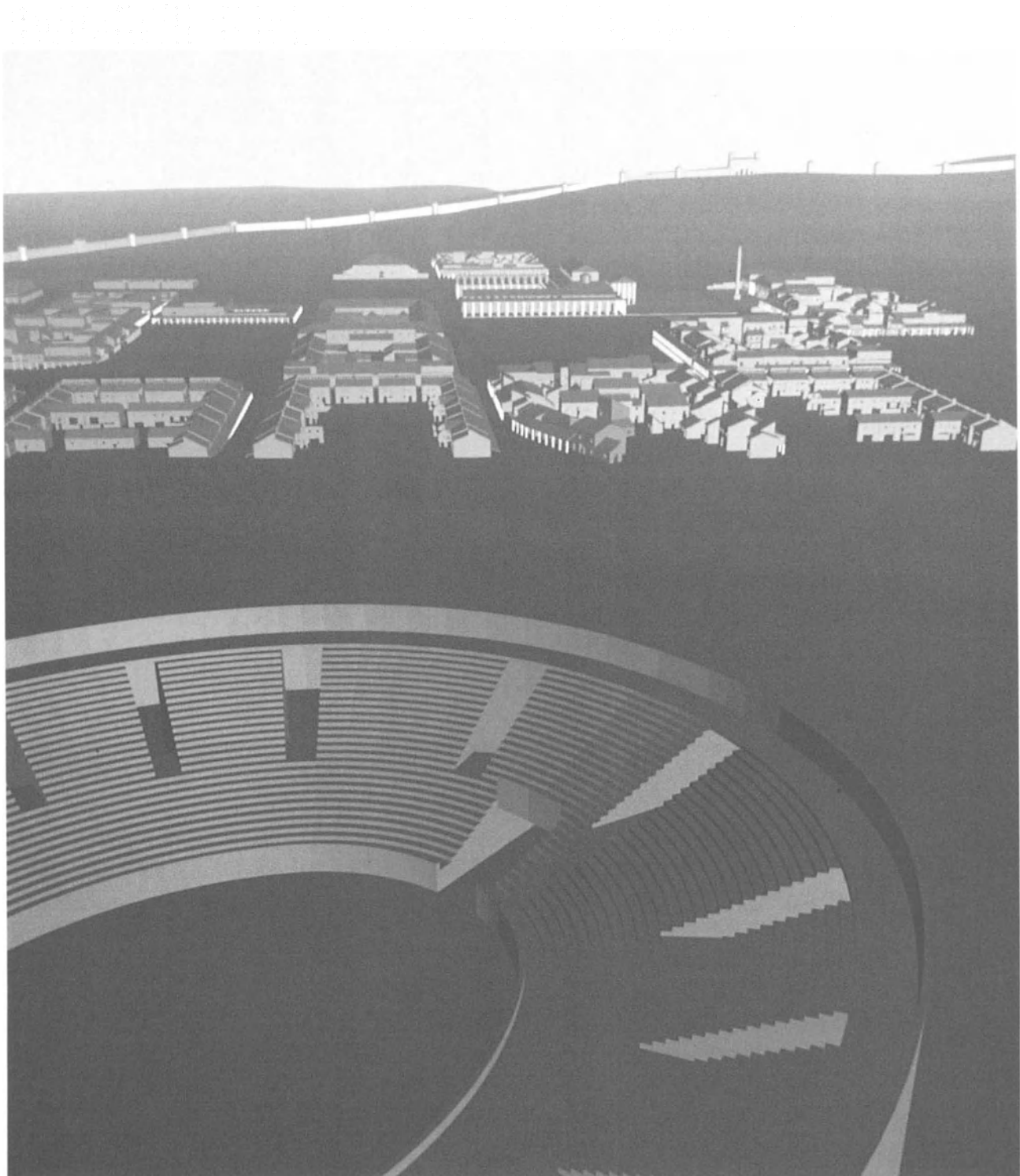
Den größten Aufschwung nahm Aventicum zu Beginn des 2. Jahrhunderts. In trajanischer Zeit (98-117) begann der Bau des Cigogniertempels, des Theaters und der Weiterausbau der Stadtmauer. Andere Bauten, wie das Amphitheater, Aquädukte und Thermen, entstanden ebenfalls im 2. Jahrhundert. Ein etwa 800 m langer Kanal und ein neuer Hafen in der Nähe der Stadtmauer erlaubten den Antransport des nötigen Baumaterials. Das Forum wurde im Schnittpunkt der beiden Hauptachsen *Decumanus Maximus* und *Cardo Maximus* angelegt. Die Tempelanlagen, Thermen und Inneneinrichtungen wurden immer

reicher. Die Stadt zählte damals über 20'000 Einwohner.

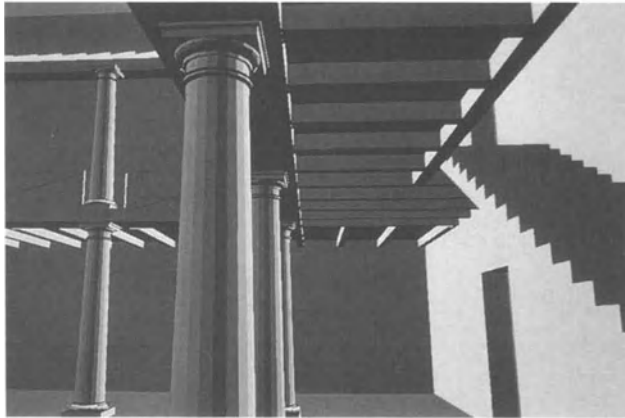
Um die Mitte des 3. Jahrhunderts begannen die Alemanneneinfälle. In aller Eile wurde auf dem stadtnahen Bois-de-Châtel ein *Castrum*, also eine kleine Festung, errichtet; die Ruinen Aventicums lieferten hierzu das Baumaterial. Nach dem Rückzug sämtlicher römischer Truppen südlich der Alpen im Jahre 401 war die gallo-römische Bevölkerung praktisch auf sich allein gestellt. Aventicum als Kleinstadt in Grenznähe verlor seine Vorrangstellung. Das heutige Avenches entstand auf einem bisher unbebauten Hügel innerhalb eines Mauerrings des 13. Jahrhunderts.

Ausgehend von dieser geschichtlichen Situation planten wir eine abstrakte Rekonstruktion Aventicums. Abstrakt deshalb, da wegen der zeitlichen Begrenzung von einem Semester eine streng wissenschaftliche und vollständige Rekonstruktion nicht in Frage kam. Es war nicht Ziel des Kurses, eine exakte archäologische und historische Rekonstruktion zu versuchen. Dazu sind Fachleute auf diesem Gebiet weitaus kompetenter [Boegli 1984]. Vielmehr bestand die Aufgabe darin, die antiken Bauherren und Architekten zu verstehen und eine architektonische Rekonstruktion vorzuschlagen. Dazu setzten wir neue Methoden ein, in denen der Modellcharakter des Entwurfs im Mittelpunkt stand. Die wenigsten Studentinnen und Studenten besaßen CAD-Erfahrungen, weshalb am Anfang eine intensive zweiwöchige Phase der Beschäftigung mit Hardware, Software und dem Netzwerk stand. Es folgte eine vierwöchige Einführung in Datenbankkonzepte, dreidimensionale Elemente, in die Instrumente der Parametrisierung, der Formengrammatiken und der Fraktale, sowie in die Methode der Prototypen.

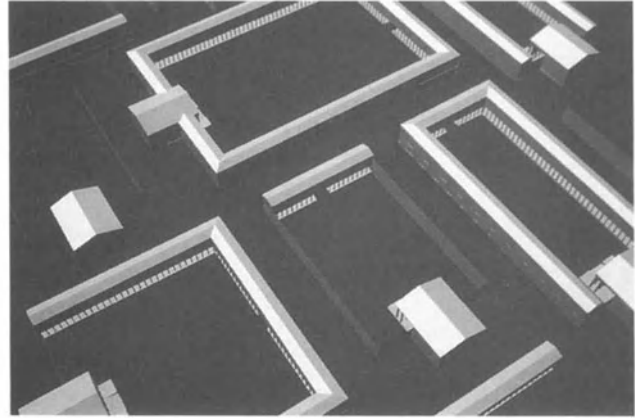
Photorealistisches Rendering der antiken Stadt Aventicum.
Amphitheater, Forum, Insulae und Stadtmauer. Van der Mark, 1991



Rekonstruktion einer Ecke des Marktplatzes, generiert mit parametrisierten Elementen. Zeidler, 1990



Beispiele für Tempel, generiert mit Tempelprototypen. Engeli, 1990



Erst danach konnte, ausgehend von einfachen Räumen bis hin zur Rekonstruktion des gesamten Forums, die eigentliche Rekonstruktionsarbeit beginnen. Die Studentinnen und Studenten wählten jeweils eine der über 40 im Stadtgebiet vorhandenen Insulae und bearbeiteten diese individuell. Das auf vernetzten Workstations installierte CAD-Programm gestattete durch das wahlweise Zuladen anderer Insulae entweder eine Gesamtschau der Stadt oder einzelner Projekte im Kontext ihrer Umgebung. Bei der Rekonstruktion konnten sich die Studierenden nahe an die historischen Vorlagen halten. Zu Beginn standen die Digitalisierung der noch vorhandenen Reste der Stadt, der Umgebung, sowie der wichtigsten archäologischen Funde. Danach begann die Anwendung der einzelnen Methoden und Instrumente.

(1) Datenbanken existierender Elemente. Bei jeder Ausgrabung werden Artefakte zutage gefördert, deren geometrische und nicht-geometrische Eigenschaften einmalig sind. Für solche Objekte bietet sich die Ablage in einer Datenbank dreidimensionaler Objekte an, ein Vorgehen, das inzwischen auch in der archäologischen Praxis zum Standard gehört. AutoCAD-

Blocks, versehen mit den entsprechenden Attributen, liefern dazu die entsprechende Speicherstruktur. Die Elemente sind visuell über Pull-Down-Menüs oder Pop-Up-Slides zugänglich. Typische Beispiele sind Möbel, römische Ziegel und Säulen. Studentinnen und Studenten konnten so ausgegrabene oder dokumentierte römische Gegenstände direkt aus einer entsprechenden Datenbank abrufen und in ihr Modell einsetzen. Dies entspricht dem Arbeiten mit einem 'Kit-of-Parts', mit dem Teile unverändert übernommen werden müssen, aber in beliebiger Zahl und Position kombinierbar sind. Eine ähnliche Option bieten die meisten kommerziellen CAD-Programme heute für Standardelemente.

(2) Datenbanken parametrisierter Elemente. Objekte unterscheiden sich oft nur durch wenige Eigenschaften. Kann man diese identifizieren und die Grenzen bestimmen, innerhalb derer die Variation stattfindet, ist ein Parameter erkannt. Parameter können die Geometrie, die Farbe, das oder sonstige Eigenschaften sein. So finden sich Säulen, deren Höhe innerhalb gegebener Grenzen variiert, deren Proportionen jedoch konstant bleiben, oder solche, die bei kon-

Blick auf Wohnbauten und auf das
Forum. Van der Mark, 1991



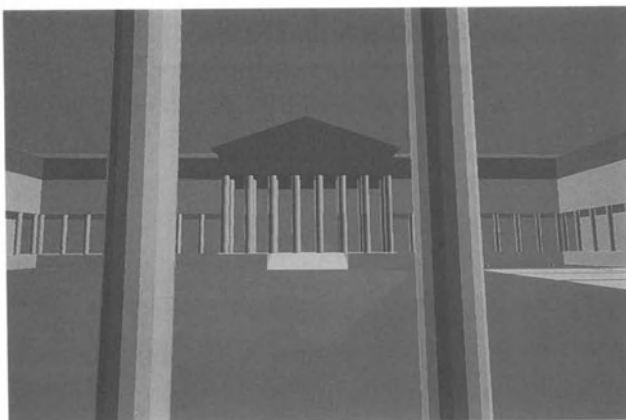
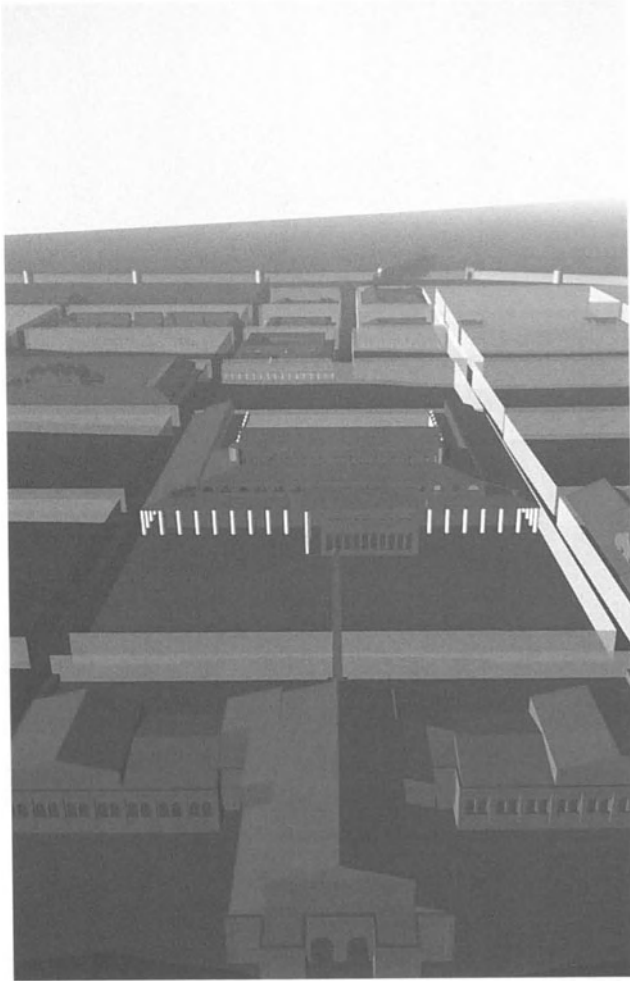
stanter Höhe verschiedene Proportionen aufweisen. Ein anderes Beispiel sind runde Steintische, deren Durchmesser innerhalb durch das Material bestimmter Grenzen variieren kann. In der ebenfalls auf AutoCAD-Blocks basierenden Datenbank befinden sich daher Elemente, deren Topologie bestimmt und deren endgültige Geometrie durch einen oder mehrere Parameter definierbar ist. Den Werten der Parameter sind Grenzen gesetzt, die mit den Objekten gespeichert sind. Die Parameter und ihre Grenzwerte wurden über modifizierte Pop-Up-Menüs eingegeben. Weitere parametrisierte Elemente waren Treppen, Geländer und Balken. Datenbanken parametrisierter

Elemente finden sich ebenfalls bereits in einigen kommerziellen CAD-Systemen.

(3) Regelbasierte Systeme und fraktale Algorithmen. In dieser Übung untersuchten wir zum erstenmal die praktische Anwendung der Instrumente Formengrammatiken und Fraktale. Flemming [Flemming 1986] und Mitchell [Mitchell 1990a] beschreiben deren Verwendung im Entwurf. Shape Grammars erwiesen sich als eine effiziente Hilfe bei der Konstruktion von geometrisch definierbaren Objekten, anwendbar in der Modellierung von Treppen bis hin zur Rekonstruktion mittelalterlicher Kathedralen oder

Oben: Visualisierte Grenzen der Insulae. Van der Mark, 1991

Unten: Blick vom Tempelgang.
Van der Mark, 1990



moderner Architekturbeispiele. Fraktale Algorithmen sind unverzichtbar in der Modellierung natürlicher Elemente, wie Landschaften und Bäume. Ihre Anwendung für die Rekonstruktion bedingt allerdings, daß die dem Aufbau zugrunde liegenden Regeln erkannt, isoliert, und durch entsprechende Mechanismen verknüpft werden. Der Prozeß der Zerlegung eines Objektes in Rekonstruktionsregeln ist dann nicht umkehrbar, wenn auch nur die kleinste Abweichung in den Regeln oder den Verknüpfungen zugelassen wird. Allerdings ist gerade diese Tatsache in der Generierung neuer Objekte aus bekannten Regeln interessant.

(4) Relationale Prototypen. Damit bezeichnen wir zwei- und dreidimensionale funktionale Prototypen von Gebäuden, die bereits einiges Wissen über sich selbst besitzen. Beispiel ist der relationale Prototyp einer römischen Therme, der weiß, welche Räume zu diesem Gebäudetyp gehören, welche Proportionen und Dimensionen die Räume haben dürfen und welche Relationen bestehen müssen oder dürfen. Während des interaktiven Entwurfs gibt das Programm Kommentare, falls gegen die funktionalen Regeln verstoßen wird.

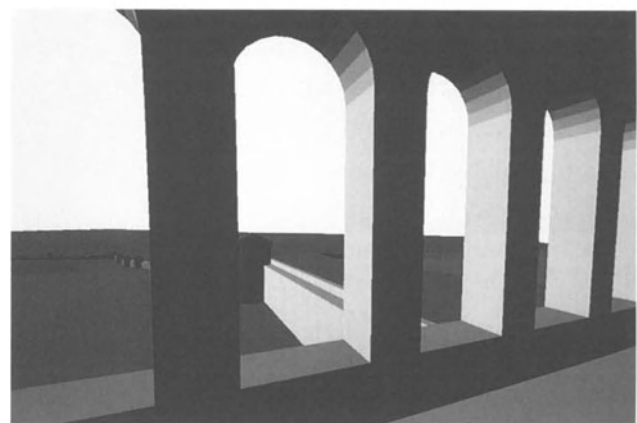
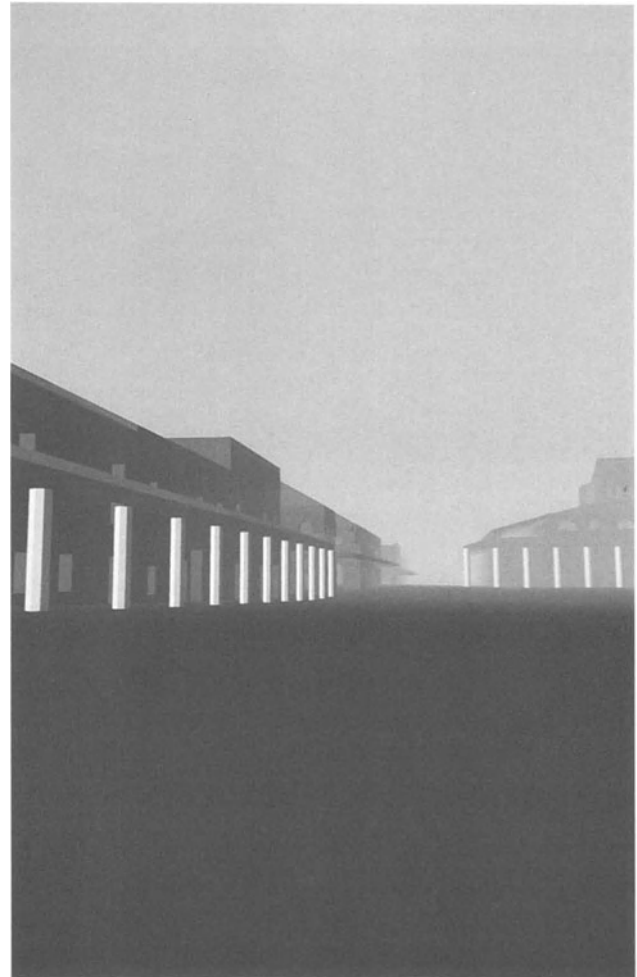
(5) Intelligente Prototypen. Prototypen kombinieren die Merkmale aller bisher beschriebenen Methoden. Sie sind parametrisiert. Die Parameter bewegen sich innerhalb gegebener Grenzen und stehen untereinander durch definierte Relationen in Verbindung. Die Änderung eines Parameters kann eine ganze Reihe Änderungen anderer Parameter nach sich ziehen. Beispiele sind römische Tempel oder Theater, die sich innerhalb eines Typs durch viele Gemeinsamkeiten auszeichnen. Im Fall des Tempel-Proto-

Oben: Blick auf die Stadt vom Südtor.
 Van der Mark, 1990

Unten: Blick in das Forum.
 Van der Mark, 1991

typs wählt der Anwender die Art des Tempels zunächst aus und bestimmt danach alle Parameter graphisch auf dem Bildschirm. Der Speicherbedarf für Objekte, die nach den einzelnen Methoden erzeugt werden, ist sehr unterschiedlich. Datenbanken individueller 3D-Objekte wachsen sehr schnell an. Bei parametrisierten 3D-Objekten ist die Topologie nur einmal gespeichert, für alle folgenden sind lediglich die Werte der Parameter verschieden. Noch kompakter ist die Speicherung in Form von wenigen Entwurfsregeln und deren Verknüpfung. Prototypen, als Objekte im Sinn des objektorientierten Programmierens zusammen mit den dazugehörenden Operatoren gespeichert, sind ebenfalls nicht sehr speicherintensiv. Sie sind der menschlichen Vorstellung von Objekten möglicherweise nahe verwandt und daher sogar in Quellcodeform leichter verständlich als lange Listen unstrukturierter Koordinatenpaare.

Alle Methoden waren innerhalb eines kommerziellen CAD-Programms abrufbar und frei kombinierbar. Sie bildeten keine inkompatiblen Inseln, sondern konnten jederzeit miteinander verbunden werden. Mit wachsender Komplexität der Operationen und Objekte stießen die Studentinnen und Studenten schnell an die Grenzen der Hardwarekapazität. Mit Absicht wählten wir für den Kurs ein historisches Thema mit Schwergewicht auf die Rekonstruktion innerhalb fester Grenzen, da hier die mit dem Computer möglichen Ansätze besonders zum Tragen kommen konnten. Für neuartige Probleme wäre eine Beschränkung auf die oben beschriebenen Methoden sehr fraglich. Sie sind nützlich bei der Lösung von wohl definierten Teilproblemen im Entwurf und zeigen, daß Simulation nicht nur für zukünftige, sondern auch für vergangene Architektur Bedeutung hat.

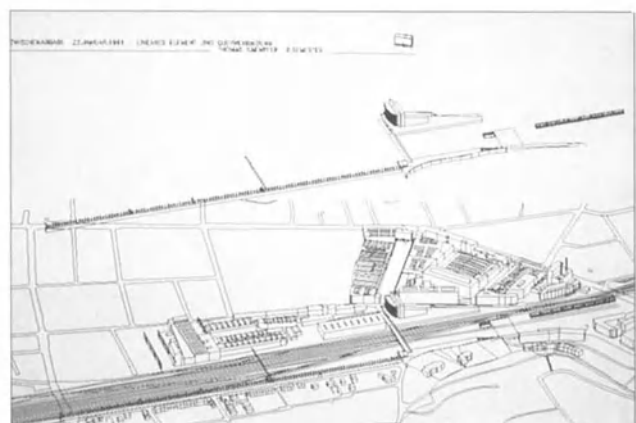
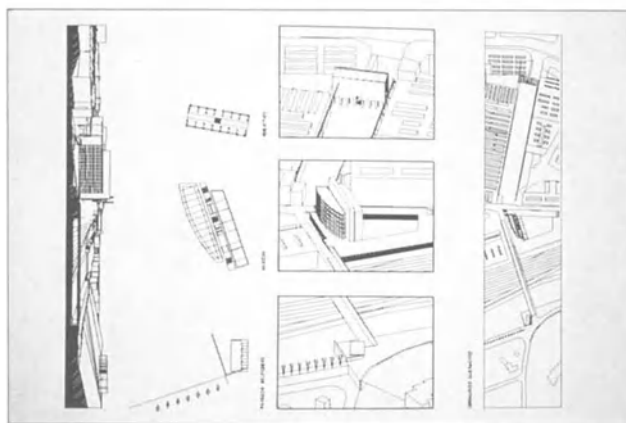


Umnutzung eines Industrieareals: Winterthur

Oben: Massenmodelle. Hirschberg, 1990

Unten: Thematische Darstellungen und

Entwicklung von Alternativen. Kämpfer, 1991



In Zusammenarbeit mit dem Architekten Mario Campi fand 1990 ein Computer Aided Architectural Design-Studio für Studentinnen und Studenten in den beiden letzten Studienjahren statt. 16 Teilnehmerinnen und Teilnehmer, zum Teil mit zum Teil ohne Computererfahrung, wollten an dem Experiment teilnehmen. Die Entwurfsaufgabe - das Industrie-Areal der Firma Sulzer in Winterthur und dessen Überführung in eine neue Nutzung und architektonische Gestalt - war komplex. In einem ersten Schritt, der gleichzeitig als Einführung in das CAD-Programm diente, digitalisierten die Assistentinnen und Assistenten die bestehende Stadt.

Wenig später erhielten wir das digitale Geländemodell der Stadt vom Vermessungsamt. Verschiedene Gebäude, Bauzustände und Funktionen wurden auf die entsprechenden Schichten (Layer) abgelegt. So entstand eine dreidimensionale, thematische Entwurfsumgebung. Die Studentinnen und Studenten wählten danach einen Teil des Geländes für ihren Entwurf aus und begannen mit dreidimensionalen Skizzen. Dabei kamen die ersten Versionen des objektorientierten Entwurfsinstruments zum Einsatz, das die Erzeugung von Variationen eines Typs und die Substitution von Entwurfsvorschlägen erlaubte.

Winterthur, im Vordergrund links das Industrieareal.
Mitarbeiterinnen und Mitarbeiter der Professuren
Campi und Schmitt, 1990



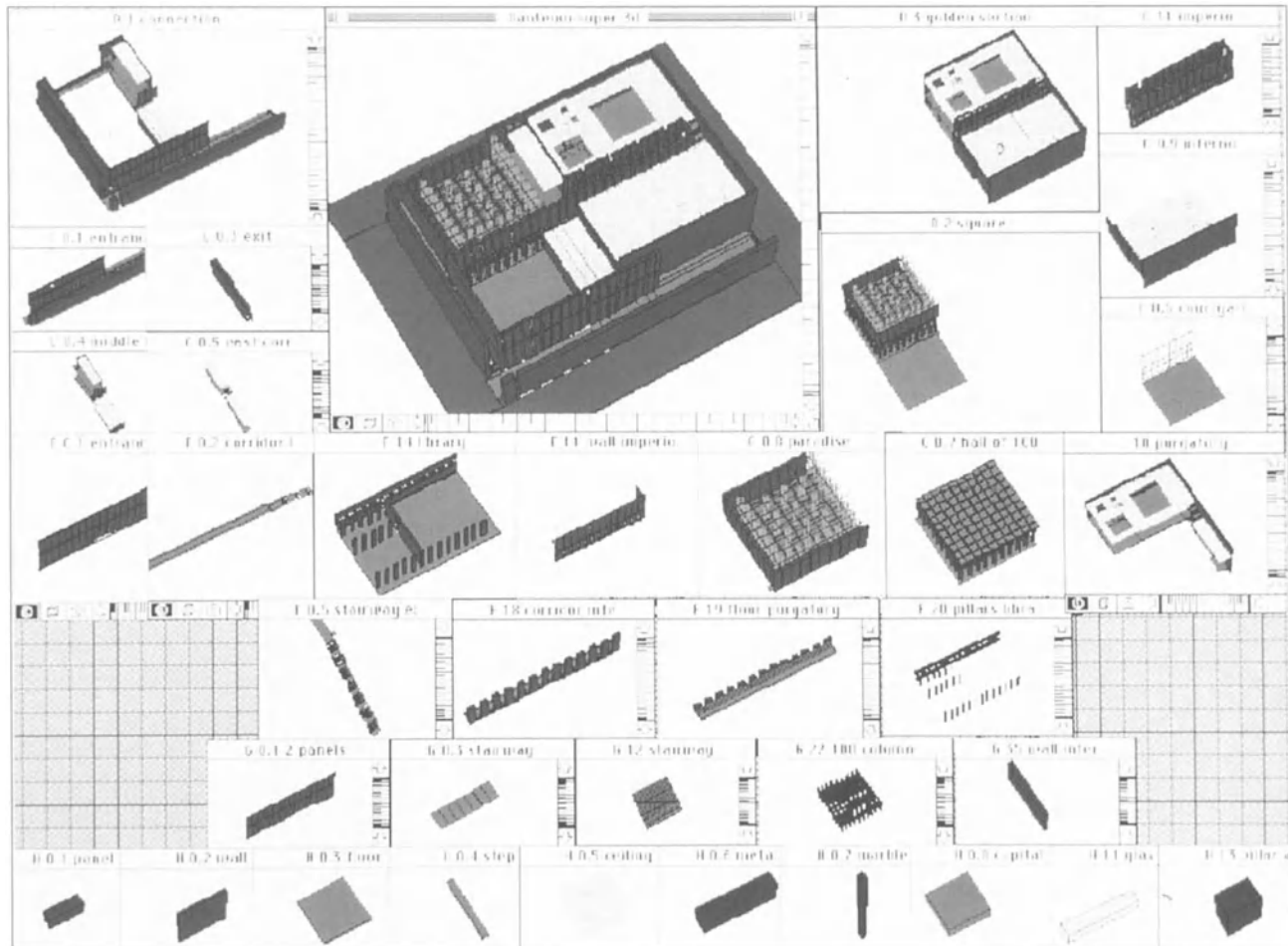
Dieses Studio zeigte uns viele Verbesserungsnotwendigkeiten der neuen Instrumente. Doch waren einige der Teilnehmer am Entwurf in der Lage, das damals noch primitive Instrument bereits effizient einzusetzen und die Entwicklung ihrer Idee überzeugend zu dokumentieren. Die so geschaffenen Modelle und Zeichnungen kamen schließlich bei einer Ausstellung über das Industrieareal Winterthur zusammen mit den traditionell hergestellten Entwürfen zum Einsatz. Zwei Studenten vertieften ihre Arbeit mit der Produktion eines Videos, das einen virtuellen Spaziergang durch den zentralen Teil des Industrieareals darstellte. Das Video von

Thomas Kämpfer zeigte auch ein neu entworfenes Museum, zu dem man über eine Brücke gelangt. Besonders interessant an diesem Video war, daß fehlende optische Details durch die geschickte Wahl akustischer Eindrücke ergänzt wurden. So definierte zum Beispiel die Kombination von Akustik und Geometriemodell eindeutig die Beschaffenheit des Bodenbelags (Kies im Innenhof) oder die Funktion eines Raumes (Café). Bei der Vorstellung der Zwischenergebnisse am Bildschirm wurden die Schwächen dieser Darstellungsart im Verhältnis zu Papierpräsentationen deutlich und zeigte die Notwendigkeit der Großprojektion.

Rekonstruktion und Analyse der Architektur Giuseppe Terragnis

Nach einem Text von Antonino Saggio, La Sapienza, Rom

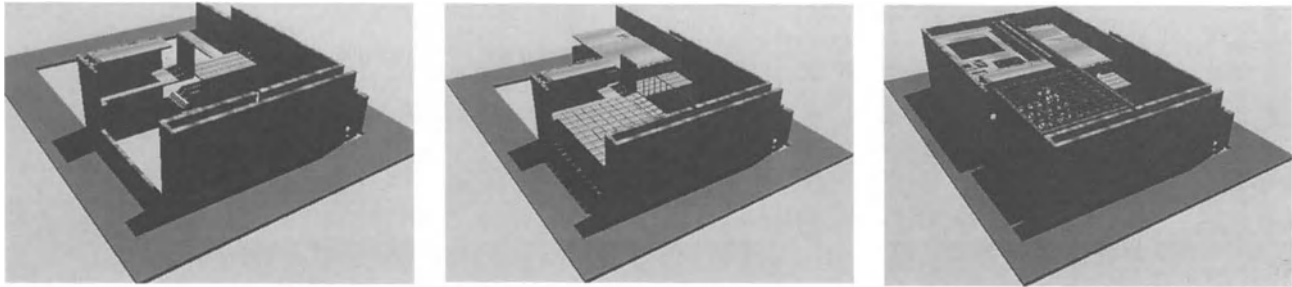
Hierarchische Zerlegung (oben) und selektive Darstellung (nächste Seite) des Danteums von Terragni und Lingeri. Zimmermann, 1992



Als Teil der Fall-Datenbank im 'Architectural Space Laboratory' der Architekturabteilung der ETH Zürich entstanden acht Modelle nicht realisierter Projekte des italienischen Architekten Giuseppe Terragni (1904-1941). Diese Modelle geben Einblick in eine wichtige Phase des italienischen Rationalismus. Die 'Accademia di Brera', zwei Versionen der Villa 'Floricoltore', das Projekt des 'Palazzo dei congressi', das 'Haus für einen Künstler', die 'Villa con Darsena', das Projekt einer Kathedrale und zusammen mit Pietro

Lingeri das 'Danteum' stehen in verschiedenen Datenformaten zur Verfügung.

Die Modelle waren das Resultat eines speziellen Lehr-Projekts mit dem Namen 'Giuseppe Terragni Architecture. A Formal Analysis Using CAAD', das Antonino Saggio im Wintersemester 1992/1993 an der ETH Zürich durchführte. Der Kurs gab den Studentinnen und Studenten die Möglichkeit, Terragnis Architektur durch Analyse und Rekonstruktion verstehen zu lernen. Zum Einsatz kamen verschiedene Modellierpro-



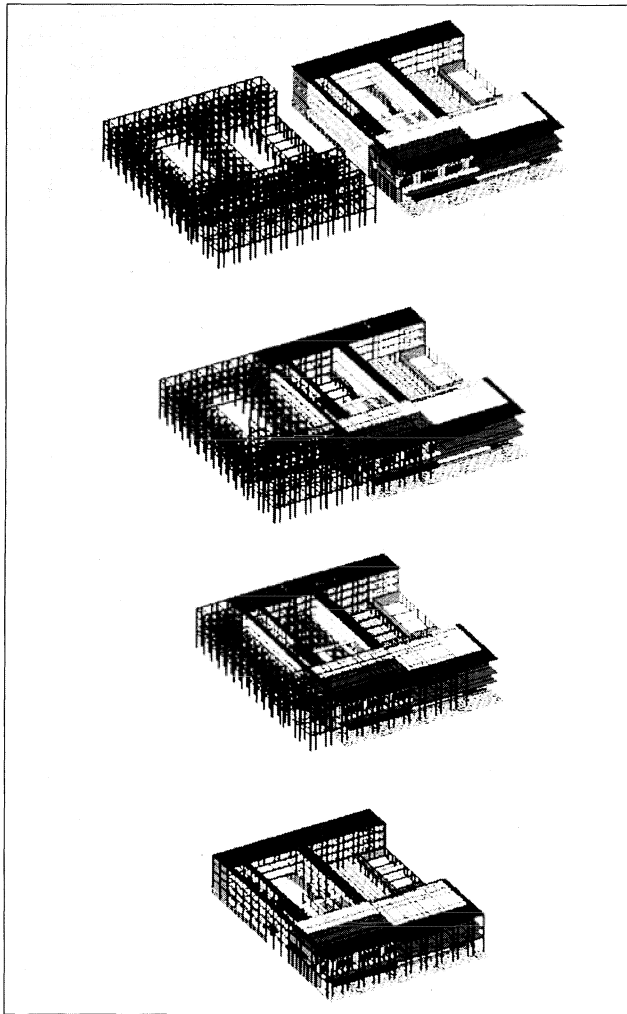
gramme für Macintosh-Computer, deren Relevanz für bestimmte Modellerschritte untersucht wurde. Nachdem die Studierenden die Grundlagen der Pixel- und Vektormanipulation, der graphischen Datenbanken, des Volumenmodellierens, der Animation und der photorealistischen Darstellung kennengelernt hatten, kamen sie zum eigentlichen Kern des Kurses, der Untersuchung hierarchischer Strukturen. Andere Anwendungen bezeichnen mit diesem Namen 'Symbole', 'Dynamische Bibliotheken', 'Typen und Variationen' und 'Objektbasiertes Modellieren'. Sie alle sind Ausdruck der für den Computer typischen dynamischen und vertikalen Organisation von Daten. Das Arbeiten mit hierarchischen Strukturen verlangt die Gliederung eines Modells in seine Teile. In einer so strukturierten Umgebung muß man die Unterschiede zwischen primitiven Objekten, zusammengesetzten Objekten, Variationen und Klassen erkennen und nutzen. Primitive Objekte sind nichts anderes als Punkte, Linien, Flächen und Volumen, die jeder dreidimensionale Modellierer zur Verfügung stellt. Zusammengesetzte Objekte sind das Ergebnis der Kombination vorgefertigter Elemente, bestehend aus primitiven Objekten, aus denen ein Hierarchie-Baum entsteht. Die verschiedenen Hierarchie-Ebenen entsprechen den Klassen im objektorientierten Programmieren.

Werden zwei Elemente zu einem zusammengesetzten Objekt verbunden, so entstehen Variationen. Während die Variationen parametrisch verformt, rotiert, kopiert und frei im Raum plaziert werden können, kann die zugrunde liegende geometrische Struktur nur auf der Ebene der ursprünglichen Objekte geändert werden. Andererseits hat jede Veränderung auf der untersten Ebene der Hierarchie die automati-

sche Veränderung in den darüberliegenden Ebenen zur Folge.

Die Modelle nicht realisierter Terragni-Projekte nutzen das Potential dieser Struktur. In allen Fällen finden sich auf der untersten Ebene der Hierarchie die primitiven Elemente, aus denen alle zusammengesetzten Objekte aufgebaut sind. Die Artikulation von Elementen und Räumen schließt die die Lücke zwischen primitiven Objekten und dem Gesamtmodell. Die Organisation ist dabei zugleich konstruktiv und semantisch, denn der Aufbau der verschiedenen Klassen zeigt die persönliche Interpretation des Entwurfs durch die Studentinnen und Studenten. Einige Modelle sind nach unterschiedlichen Funktionen aufgebaut, andere nach tragenden und nicht-tragenden Elementen, wieder andere nach linguistischen Modellen unter Schlüsselworten, wie Rahmen, Verbindungen und freie Elemente. Der Aufbau des Danteums beispielsweise basiert auf den verschiedenen Funktionen der einzelnen Räume. Das Modell besteht aus über 50 verschiedenen zusammengesetzten Objekten und ist insgesamt auf vier verschiedenen Ebenen organisiert: primitive Elemente, zusammengesetzte Elemente, funktionelle Teile, Gesamtmodell.

Die Anwendung hierarchischer Strukturen hat in der Analyse und Rekonstruktion von Architektur folgende Vorteile: (1) Sie erlauben die Konstruktion eines komplexen Modells auf einer einfachen Computer-Plattform. Die Aufteilung in verschiedene Klassen mit wiederverwendbaren Elementen minimiert den Speicheraufwand. (2) Entwurf und Manipulation der verschiedenen Teile der Hierarchie, aus denen sich das Modell zusammensetzt, werden durch Ein- und Ausblenden von Information erleichtert. Einerseits erhöht dies die Geschwindigkeit des Rendering, ande-



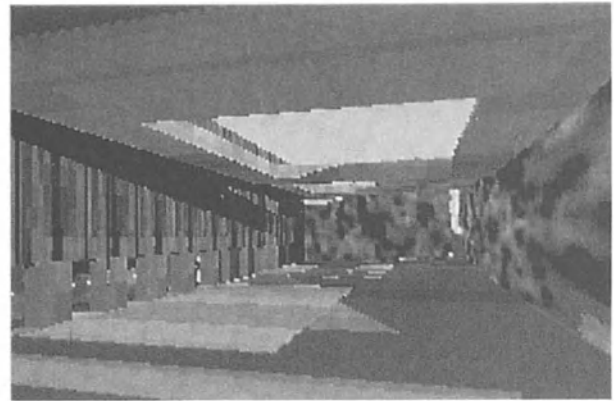
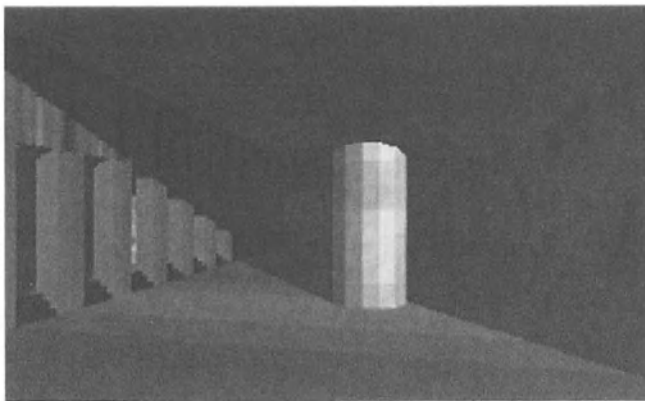
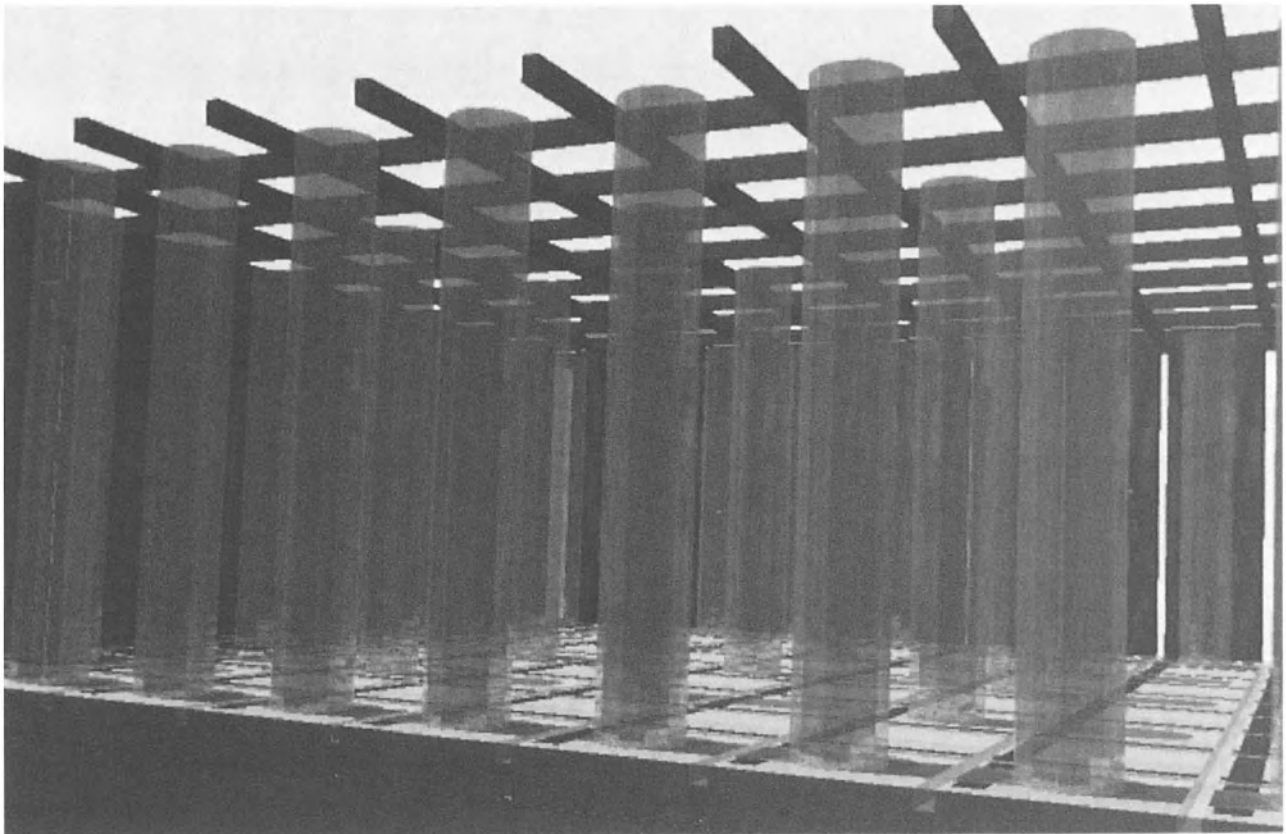
Überlagerung von Gebäudestruktur und
Gebäudehaut. Palazzo dei congressi von
Terragni und andere, 1937. Feederle, 1992

rerseits erlaubt es jederzeit die Darstellung der Interpretation des Gebäudes. So läßt sich einfach eine Reihe didaktischer Bilder erzeugen, die sich für den späteren Einbau in eine Animation oder in wichtige Zeichnungen eignen. (3) Die Artikulation einer hierarchischen Struktur gestattet die realistische Simulation und detaillierte Studie der Materialien. Da die Veränderung der Eigenschaften der Primitive, wie Farbe, Transparenz oder Glanz, sich über alle Ebenen der Hierarchie fortsetzt, ist es möglich, entsprechende Materialhypothesen photorealistisch zu untersuchen. (4) Die gesamte Modellkonstruktion kann auf einer langsamen Maschine geschehen. Die Resultate lassen sich danach auf Visualisierungsmaschinen übertragen.

Hierarchische Modell-Strukturen bieten eine Umgebung für die Repräsentation und daher auch für die

Untersuchung und Simulation von Architektur, die keine traditionelle Entsprechung kennt. Sie sind geeignet und konsistent für die Disziplin der Architektur. Im Unterschied zur traditionellen Vorgehensweise werden die Ergebnisse des Dekompositions- und des Kompositions-Prozesses nicht auf verschiedenen Papierdokumenten abgelegt; die wichtigen Ideen zur Interpretation der Architektur sind vielmehr in der Struktur des Aufbaus des Modells enthalten. Daher bedingen sich das Verstehen eines Entwurfs und seine hierarchische Konstruktion wechselseitig.

Neue Erkenntnisse dieser Modellierung basieren auf der vielfältigen Interpretationsmöglichkeit der zugrunde liegenden Strukturen. Da bekannt ist, daß Repräsentation, Simulation und Erforschung eines Projekts miteinander in Verbindung stehen, ergeben

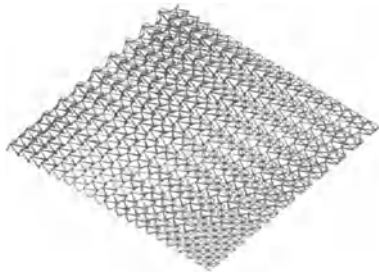


Simulierte Blicke in das Danteum: Paradies,
Inferno und Fegefeuer. Zimmermann, 1992

sich aus den Interpretationsmöglichkeiten der Strukturen auch Interpretationsmöglichkeiten der Architektur. Ein mit hierarchischen Strukturen rekonstruiertes Projekt ergibt ein anderes Produkt als eine herkömmliche Entwurfsübung. Es ist ein lebendiges, dynamisches und interaktives Modell der Wirklichkeit, das zugleich eine Struktur für das kritische und

realistische Testen bietet. Diese Charakteristika koexistieren in demselben Computermodell, das so dem Traum vom intelligenten Modell näher kommt, denn es kann mehr Wissen zeigen, als ursprünglich eingegeben wurde [Saggio 1992]. Damit entstehen neue Hypothesen und Erkenntnisse, mit denen zu Beginn nicht zu rechnen war.

CAAD-Programmentwicklung



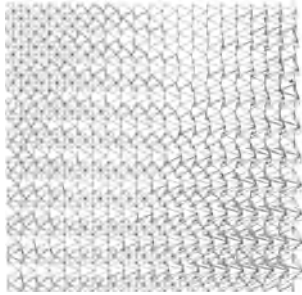
Die Vermittlung von Programmentwicklungs-Kenntnissen in der Architekturausbildung ist ein umstrittenes Thema. Die Befürworter sehen darin eine überzeugende Möglichkeit, Studentinnen und Studenten zum systematischen Denken und Planen zu erziehen. Die Gegner halten es für eine unnötige Belastung des ohnehin überfüllten Stundenplanes, die nichts mit dem Erlernen des Architektur-Schaffens zu tun habe. Ist das Programmieren erst einmal eingeführt, wird es entweder bei der Informatik gelehrt oder bei den Architekten selbst. Die Computerwissenschaftler garantieren die 'reine Lehre' des Programmierens, haben aber meist keinen Bezug zur Architektur. Die Architekten sind in der Vermittlung des Programmierens eher an der Anwendbarkeit im Entwurf interessiert. Wir haben letztere Möglichkeit gewählt. Die Anwendung aller zuvor beschriebenen Methoden und Instrumente setzt das Vorhandensein von Programmen voraus. Der Entwurf und die Erstellung solcher Programme ist eine Tätigkeit, die mit Entwurf und Ausführung in der Architektur vieles gemeinsam hat. Ganz besonders zählt sich bei der Programmentwicklung eine gute Planung und ein systematisches Vorgehen aus, denn ein nützliches Programm findet viele Anwender, die sich auf seine einwandfreie Funktion verlassen können müssen. Dementsprechend legen wir in der Einführung in die CAAD-Programmentwicklung großen Wert auf diesen Aspekt. Unsere Erfahrung zeigt, daß ein Programm sich um so besser für den kreativen Einsatz eignet, je klarer und exakter die Definition und die Ausführung sind. Für Anwenderinnen und Anwender sind die Benut-

zeroberfläche und der Inhalt des Programms selbst von größter Bedeutung. Daneben sind für das Programmieren von CAAD-Anwendungen andere Aspekte wichtig, die wir in den folgenden Schritten behandeln:

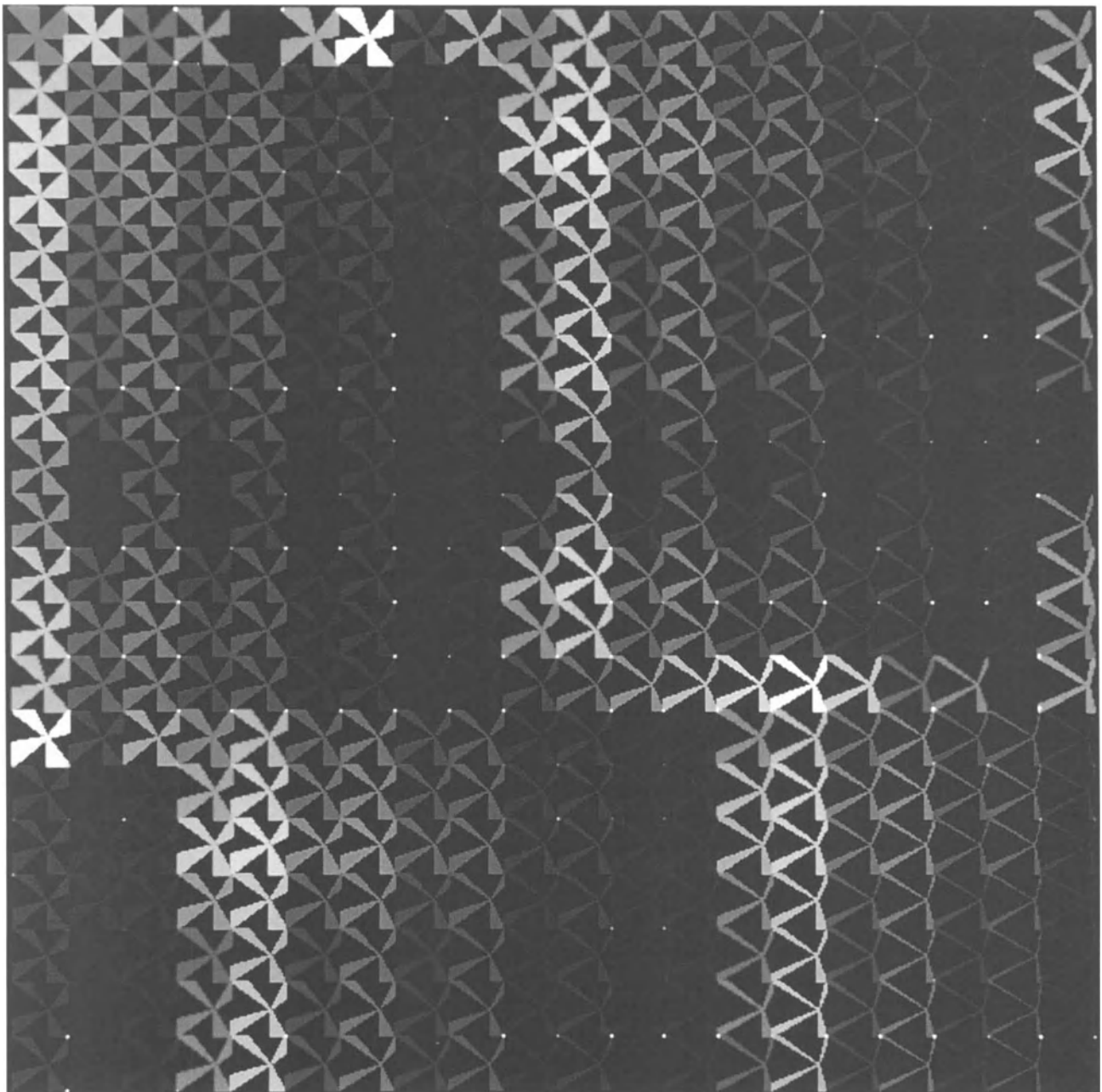
(1) Einführung in die Programmiersprache Lisp. Wir erklären und demonstrieren Grundbegriffe wie Listen (Lists) und Atome (Atoms), arithmetische Operationen, Variablen, die ersten Operationen mit Listen wie 'car' und 'cdr' und die Bedeutung von 'nil'. Kleine Übungen können direkt im Textfenster von AutoCAD mit AutoLISP ausgeführt werden. Die erste Übung umfaßt Operationen mit Listen.

(2) Einführung in das Graphikpaket. Die Verbindung zwischen Graphik und Programmieren ist uns sehr wichtig. Deshalb steht früh die Erklärung der Arbeitsweise und Kontrolle des Graphikprogramms im Mittelpunkt. Das CAD-Programm ist für uns lediglich eine Graphik-Maschine, die wir mit den eigenen Lisp-Programmen steuern. Wir beginnen mit den wichtigsten Konstruktionsbefehlen und den Möglichkeiten des Editierens. Danach konzentrieren wir uns auf die Operationen mit Dateien und auf die Kontrolle des Bildschirms. Inhalt der zweiten Übung ist das Zusammenstellen einfacher Makrodateien.

(3) Prädikate und Funktionen. Wahr (True) und falsch (False), logische und Boolesche Operationen bilden den Anfang. Danach werden die Lisp-Funktionen 'cons', 'append', 'list' und 'defun' behandelt. Inhalt der dritten Übung sind weitere Operationen mit Listen, um das Arbeiten mit dieser Grundstruktur von Lisp ganz zu beherrschen.



Parametrische Formgenerierung: Wenige
Parameter ergeben nahezu unbegrenzte
Formenvariationen. Programm Dave, 1992



(4) Funktionen und Kontrollstrukturen. Evaluation von Lisp-Ausdrücken (Expressions), Funktionen und Spezialformen (Special Forms), Konditionen (Conditionals) wie 'if-then-else' und 'cond' stehen im Mittelpunkt. Inhalt der vierten Übung ist das Erstellen eigener Lisp-Funktionen und Dateien.

(5) Iteration und Rekursion. In dieser Vorlesung behandeln wir Wiederholung (Repetition) und Iteration (Iteration) mit 'while', 'repeat' und 'foreach'. Es folgt die Einführung in das Prinzip der Rekursion (Recursion). Inhalt der fünften Übung ist ein Programm, das Ornamente oder gotische Kathedralenfenster erzeugt.

(6) Datenbank und Kontrolle. Diese Vorlesung behandelt das Datenbank-Management in AutoCAD und die Entwicklung einer eigenen Datenbank innerhalb von AutoCAD. Dazu erklären wir die Möglichkeiten der Assoziationsliste (Association List), bestehend aus deren Definition, den Zugriff darauf und ihre Änderung. Als weitere Möglichkeit werden 'dotted pairs' und die 'cons' Funktion vorgestellt. Inhalt der sechsten Übung ist die Zusammenstellung eines graphischen Vokabulars und seine Ablage zur weiteren Verwendung in einer Datenbank.

(7) Zugriff auf geometrische Einheiten und auf die System-Umgebung. Anstatt eigene Funktionen zu programmieren, lehren wir in dieser Vorlesung die Nutzung von Funktionen im CAD-Programm, die den Zugriff auf die Konstruktions-Objekte erlauben. Von besonderem Interesse ist dabei der Zugriff auf den Objekt-Namen (Entity Name), die Objekt-Daten (Entity Data), das Auswahl-Set (Selection Set) und die Symbol-Tabelle (Symbol Table). Zudem lernen die Studierenden den Zugriff auf die verschiedenen Ein- und Ausgabe-Geräte (Devices). Inhalt der siebten Übung ist die Herstellung eigener graphischer Menüs (Pull-Down Menüs)

(8) Parametrisierte Formen. Die Studentinnen und Studenten sind nach der Theorie jetzt in der Lage, Rhythmus, Proportion, Symmetrie und dimensionslose Repräsentation (dimensionless Representation) zu verstehen und anzuwenden. Inhalt der achten Übung ist

die Programmierung parametrisierter Objekte, deren Parameter interaktiv zu manipulieren sind.

(9) Fraktale. Als Einführung in die Welt der Grammatiken stellen wir hier die wichtigsten Prinzipien fraktaler Formen und Algorithmen vor. Die Studierenden lernen die Programmierung der Koch-Kurve, der C-Kurve und die Bedeutung der fraktalen Dimension. Inhalt der neunten Übung ist die Programmierung einer fraktalen Kurve.

(10) Formengrammatiken. Das graphische Pendant zu den Produktionssystemen sind die Formengrammatiken. Die Studierenden lernen deren Anwendung auf geometrische Kompositionen und in der Architektur. Inhalt der zehnten Übung ist das Programmieren einer einfachen dreidimensionalen Block-Welt (Blocks World), in der die Benutzer interaktiv nach Regeln Blöcke komponieren können.

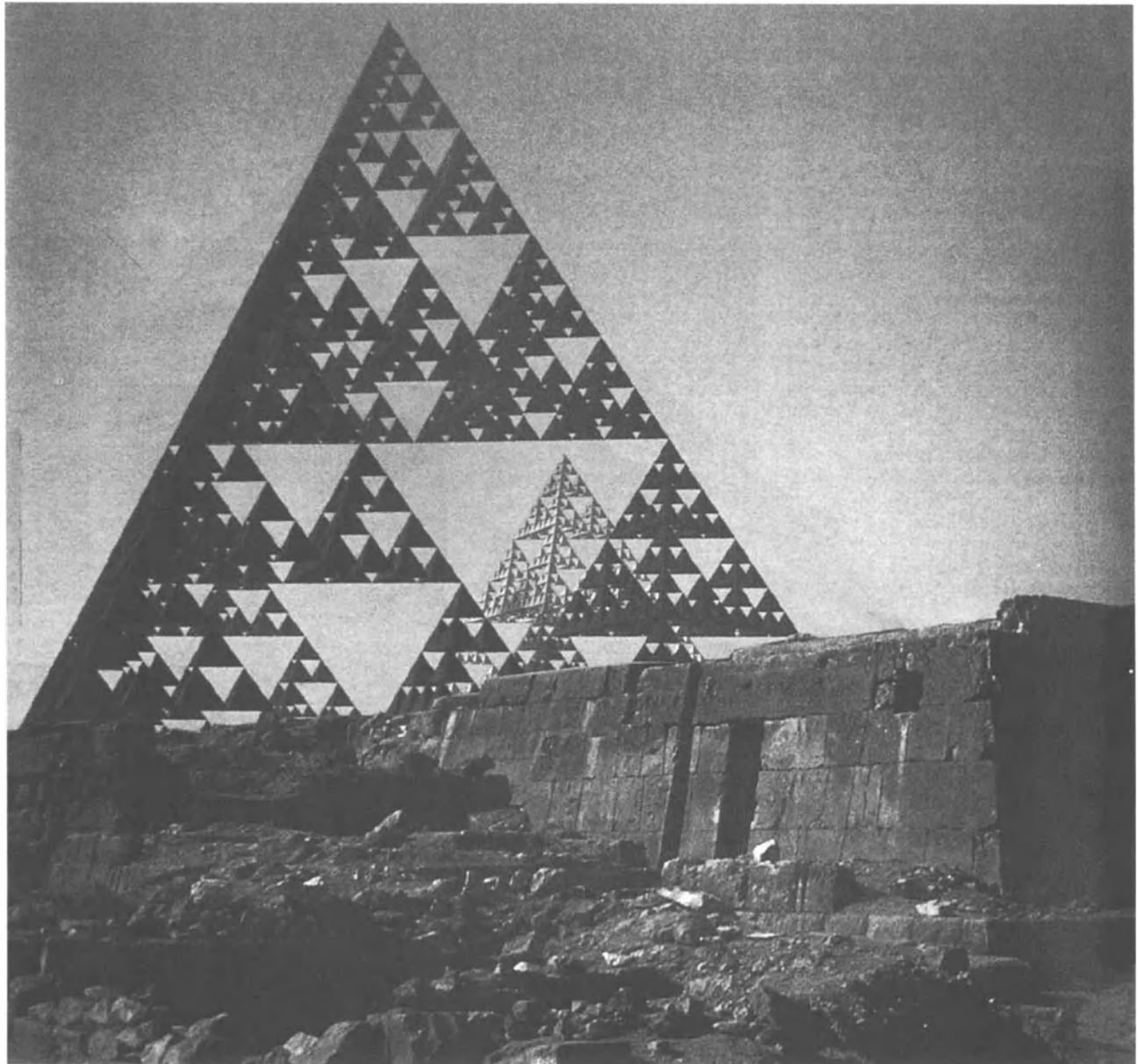
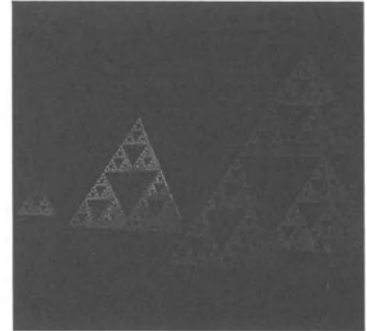
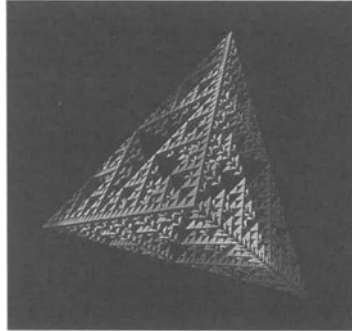
(11) Programmiersprachen. Nach dem Erlernen einer Programmiersprache (Lisp) werden die Prinzipien weiterer Sprachen wie C, Modula und Smalltalk erklärt. 'Data Typing', der Unterschied zwischen interpretierten (interpreted) und kompilierten (compiled) Programmen, sowie zwischen symbolischen und prozeduralen Sprachen stehen im Mittelpunkt. Inhalt der elften Übung ist die Kompilierung eines einfachen C-Programms und Vergleiche im Verhalten mit interpretierten Lisp-Programmen.

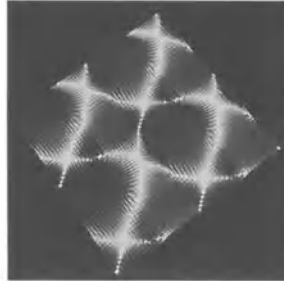
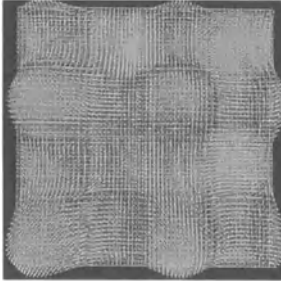
(12) Visualisierung. Die Studierenden lernen die Zusammenhänge zwischen Programmieren und Visualisieren kennen. Inhalt der zwölften Übung ist die Exploration eines zuvor erzeugten Objekts im virtuellen Raum.

(13) Abschlußpräsentation. Für die letzte Stunde stellen die Studentinnen und Studenten ihre Programme über eine graphische Oberfläche abrufbar zusammen und demonstrieren die Funktion ausgewählter Übungen. Dabei kommen oft erstaunliche Erfindungen zutage.

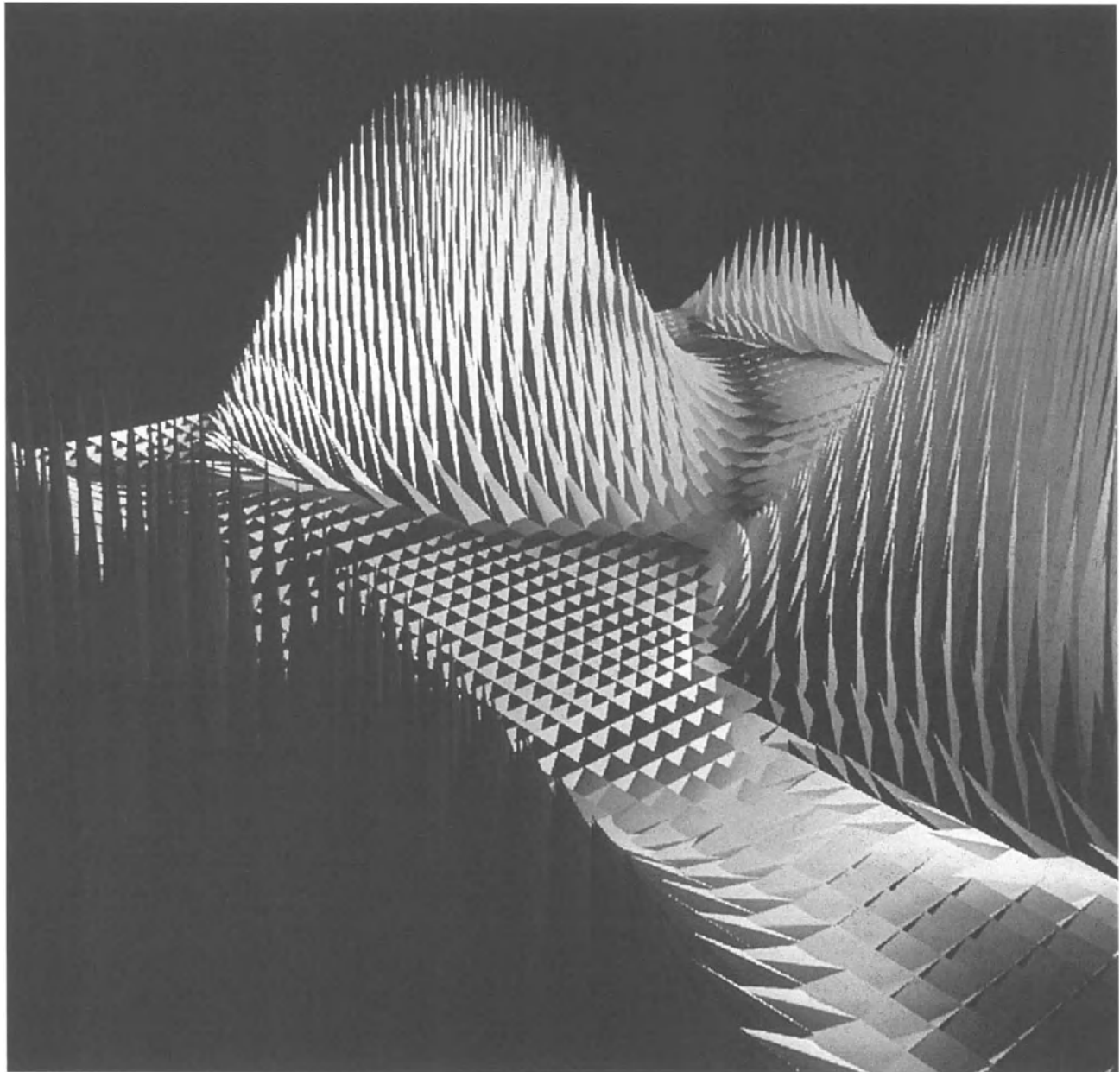
Als Grundlage des Programmierkurses dienen ein Skript [Dave 1992b] und Programmier-Referenzen [Brooks 1985], [Hancock 1986]. Jede der Vorlesungen und Übungen behandelt einen Einzelaspekt.

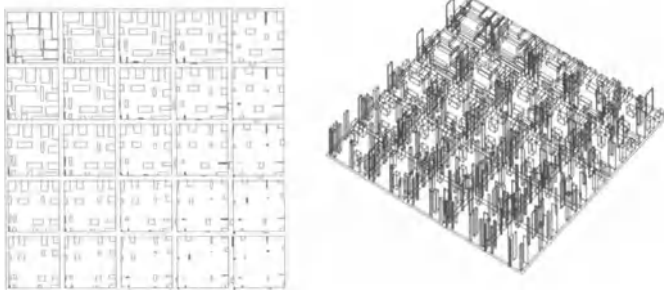
Fraktale Pyramide, aufgebaut aus
räumlichen Sierpinsky-Dreiecken.
Programmierung 9: Fraktale. Fleischli,
1993, Visualisierung van der Mark, 1993



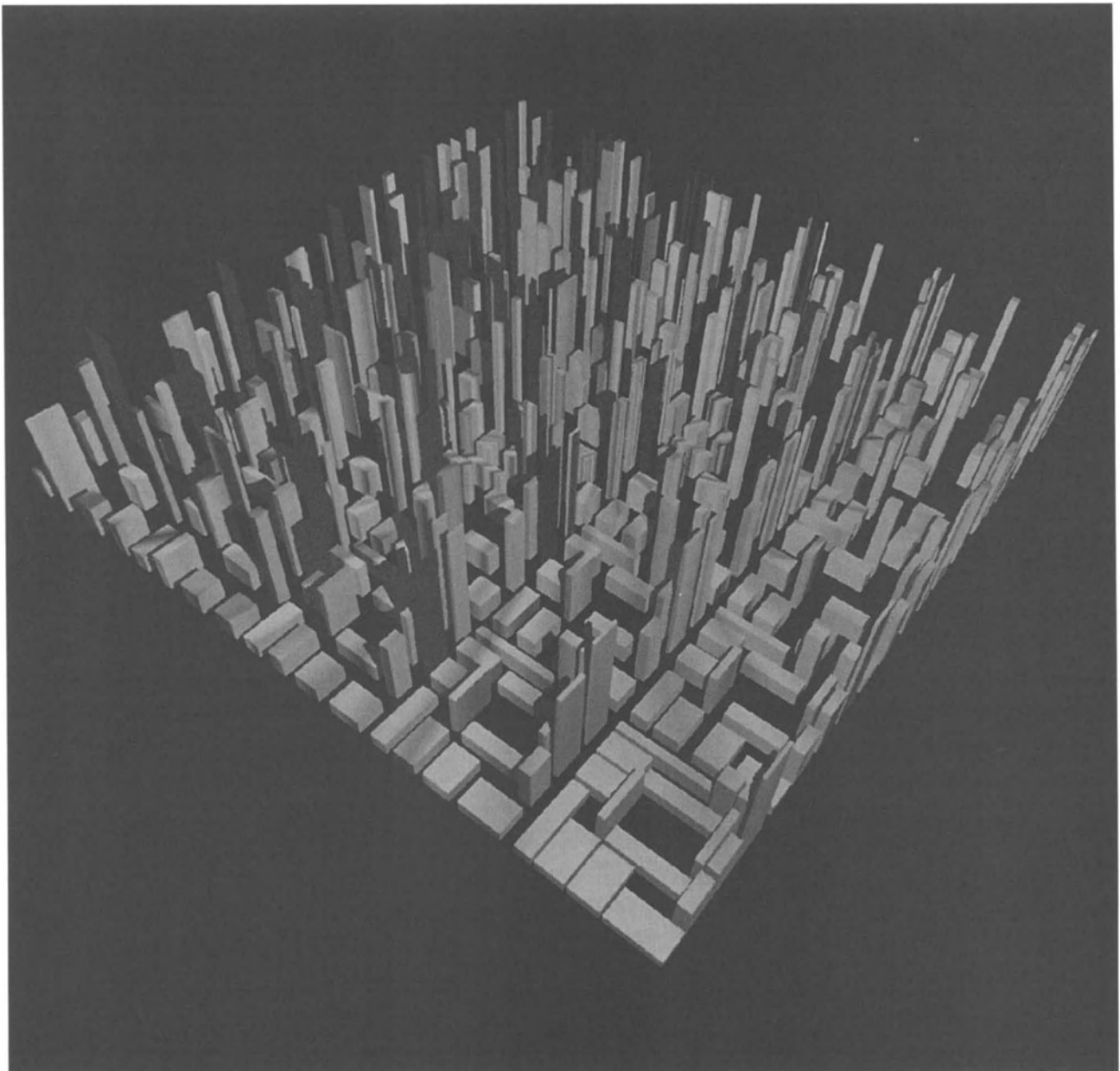


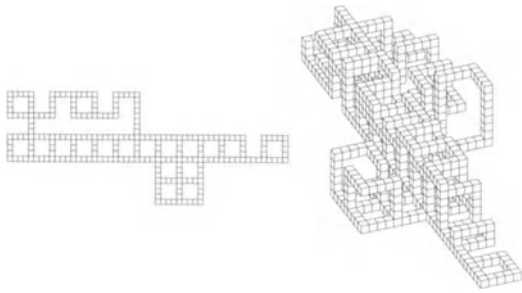
Beispiel zur Programmierübung 8: Parametrisierte Formen.
Fleischli, 1993, Visualisierung Schmitt, van der Mark, 1993



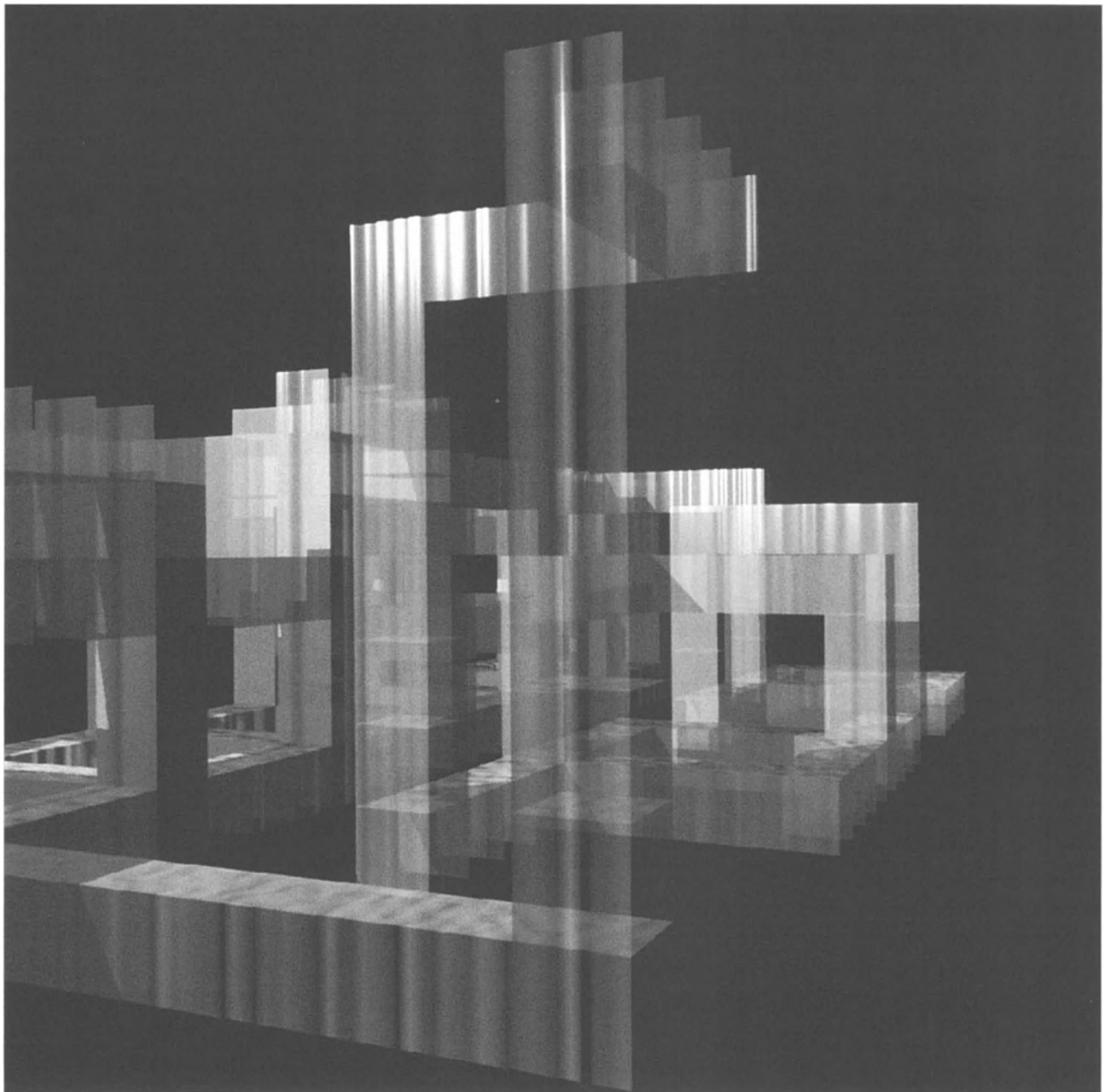


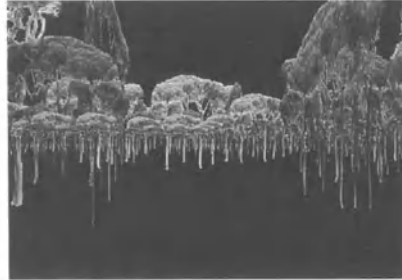
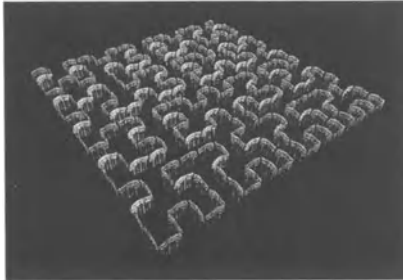
Beispiel zur Programmierübung 8: Parametrisierte Formen. Summerauer 1993, Visualisierung Schmitt, van der Mark, 1993





Beispiel zur Programmierübung 10:Formengrammatiken.
Summerauer, 1993, Visualisierung Schmitt, van der Mark, 1993





Beispiel zur Programmierübung 9: Fraktale.
Summerauer, 1993. Verwandlung der Linien
der Hilbert-Kurve in Flächen, anschließende
Belegung der Flächen mit Baummustern.
Visualisierung Schmitt, Summerauer, 1993



Praxis und Machina

Der Beginn der neunziger Jahre brachte eine explosionsartige Verbreitung der neuen Technologie in bekannten Architekturbüros Europas. Programme und Maschinen haben einen Grad der Zuverlässigkeit erreicht, der ihre Verwendung in Großprojekten erlaubt. Die folgenden Beschreibungen sind unvollständig und schlaglichtartig und können lediglich die Situation zu einem bestimmten Zeitpunkt beschreiben. Doch geben sie aufschlußreiche Einblicke in die Gründe und Denkvorgänge, die zur Computerisierung der Büros führten.

Gemeinsam ist allen ausgewählten Architekten eine der Zukunft aufgeschlossene Haltung und der jegliche Ideologie und Computergläubigkeit ablehnende pragmatische Einsatz eines neuen Architektur-Instruments. Sie streben, wie Jean-Pierre Cousin es stellvertretend für alle ausdrückt, nach einem einzigen Ziel: "garder le leadership absolu de la maîtrise d'oeuvre" [Cousin 1992]. Die angesprochenen Architekten sind der Meinung, daß sich insbesondere ihre großen Projekte ohne Computer nicht mehr realisieren ließen. Allen gemeinsam ist, daß sie die neue Technik zu einem hohen Grad eingeführt haben. Die Computerinfrastruktur ähnelt stark derjenigen an den führenden Architekturschulen Europas und ist geprägt von dezentralen, vernetzten Workstations mit Standard Software. Benutzt wird, was sich bewährt hat.

Valode et Pistre

Stellvertretend steht ein Besuch bei den Architekten Valode et Pistre am Beginn der Beschreibung. Das Büro in der Rue Renard in Paris in der Nähe des Centre Pompidou hat die Maschinen in einfacher Art integriert und sie auf die bestehenden Tische plaziert. Das Büro bearbeitet vorwiegend Großprojekte, die in Jah-

resberichten dokumentiert sind [Heylliard 1992]. Bei der Bearbeitung des Technocentre Renault, eines 600'000 m² großen Projekts auf einem 150 Hektar großen Baugebiet, stellte sich heraus, daß die Aufgabe ohne Computer nicht mehr zu lösen war. Dies führte zur Installation der ersten Workstations, denen schnell weitere folgten. Das Ziel bestand darin, alle Informationen für jedes Projekt strukturiert und spezifisch, schnell und für die Anwender transparent auszutauschen, und dabei einen hohen Grad an Datensicherheit zu gewährleisten. Die traditionellen Angebote der Berater, die entweder PC-Lösungen oder Großsysteme vorschlugen, wurden schnell verworfen. Stattdessen wurde eine Client-Server UNIX-Umgebung geschaffen.

Wie zu erwarten, ergaben sich notwendige Änderungen in der Arbeitsmethode. Von allen Mitarbeiterinnen und Mitarbeitern werden Kenntnis der grundlegenden UNIX-Befehle, prinzipielle Kenntnis des CAD-Paketes und seiner wichtigsten Befehle, sowie eine genaue Kenntnis der projektspezifischen Organisation und Prozeduren verlangt. Damit scheint das junge Team das gewaltige Bauvolumen meistern zu können. Ein Großteil der Arbeit am Computer besteht in der zweidimensionalen Konstruktion; die dreidimensionalen Fähigkeiten des Programms kommen kaum zur Anwendung. Für Präsentationen wird unter anderem das ebenfalls in Paris ansässige Büro Espace & Stratégie engagiert, das Videos und photorealistische Darstellungen der Projekte produziert.

Besonders eindrucklich zeigte sich das Potential des Computers im Entwurf am Beispiel des Projekts L'Oréal in Aulnay-sous-Bois [Heylliard 1991]. Aus einem Torus wurde ein Teil der Oberfläche in quadratischer Form ausgeschnitten. Die resultierende Figur ist elegant und mathematisch einfach bestimmbar

zugleich. Die überdeckte Halle von 33'000 m² wirkt leicht und präzise. Die Geometrie garantiert zudem eine ökonomische Konstruktion. Die Computerinfrastruktur im Dezember 1992 bestand aus fünf Macintoshes für die Verwaltung und Korrespondenz und 36 SparcStations IPX für die Architekten. Die hauptsächlich verwendete Software war AutoCAD. Die Maschinen und Digitalisieretablets befanden sich auf allen ehemaligen Zeichentischen.

Jean Nouvel

Ende 1992 befand sich das Büro von Jean Nouvel in einem ehemaligen Fabrikgebäude im Norden von Paris. Im Gegensatz zu der ruhigen Arbeitsumgebung bei Valode & Pistre herrscht ein emsiges Treiben an gefüllten Zeichentischen. Ein Schwirren liegt in der Luft, das mit der Architektur des Büros harmoniert. Man spürte das Streben nach der Realisierung einer Idee bis zu einem hohen Grad der Konsequenz.

Auch hier ist der Computereinsatz von Pragmatismus geprägt. Die Projektbezogenheit kommt stark zum Ausdruck: Für jedes neue Projekt werden einige PCs dazugekauft, die danach mit den anderen PCs vernetzt werden. Stärker als bei Valode et Pistre ist hier das dreidimensionale Modellieren Gegenstand des Interesses. Eine darauf spezialisierte Gruppe innerhalb des Büros übernimmt diese Aufgabe.

Das Büro arbeitet ebenfalls mit der Firma Espace & Stratégie, die entsprechende Videos und Renderings herstellt. Besonders für die neuen Großprojekte wie das Kaufhaus an der Friedrichstraße in Berlin kommt diese Technik zum Einsatz. Es wird klar, daß Architekt und Image-Produzent hier eine enge Verbindung eingegangen sind, denn die Ergebnisse bringen die Ideen Nouvels gut zum Ausdruck und sind vorbildlich für die

Simulation einer Vorstellung. Die Computerinfrastruktur im Dezember 1992 bestand aus Compaq-PCs mit dem Betriebssystem DOS und der CAD-Software AutoCAD, vernetzt mit Novell. Das dreidimensionale Modellieren geschah mit ARC+. Einfache Script-Dateien ermöglichen das Abspielen von Dia-Shows auf dem Bildschirm.

Ricardo Bofill

Ricardo Bofill ist mit einem repräsentativen Büro in Paris vertreten. Bereits seit einigen Jahren benutzt sein Büro den Computer zur Dokumentation und zur Produktion. Seine Architektur läßt den Schluß zu, daß Maschinen früh im Entwurfsprozeß zum Einsatz kamen, doch war dies zunächst nicht der Fall. Bereits vor Jahren wurden photorealistische Renderings und Montagen versucht, unter anderem durch die ebenfalls in Paris ansässige Gruppe Archivideo.

Das Büro modelliert mit einer Computerinfrastruktur, die aus IBM-Produkten besteht. Hardware-Plattform ist das RISC 6000-System, Software ist das von Dassault entwickelte und von IBM vertriebene Programm CATIA. Die Workstations sind miteinander vernetzt und werden zum zweidimensionalen Zeichnen und zum dreidimensionalen Modellieren benutzt.

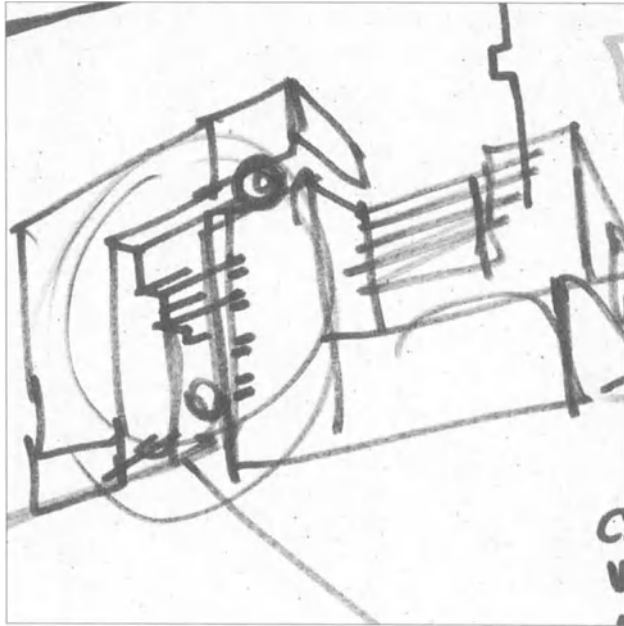
Die nächste wichtige Stufe nach der Einführung der Büroinformatik und des CAD in die Architekturbüros ist die Nutzung der digitalen Kommunikation zwischen verschiedenen Büros über Modems oder ISDN. Auch hier steht ein pragmatischer Ansatz im Mittelpunkt und die meisten Anwendungen sind auf das Austauschen von Dateien beschränkt. Mit wachsender Kompetenz und höheren Übermittlungsgeschwindigkeiten ist abzusehen, daß auch hier die neuen Möglichkeiten besser genutzt werden.

Ein Wettbewerb für Granada

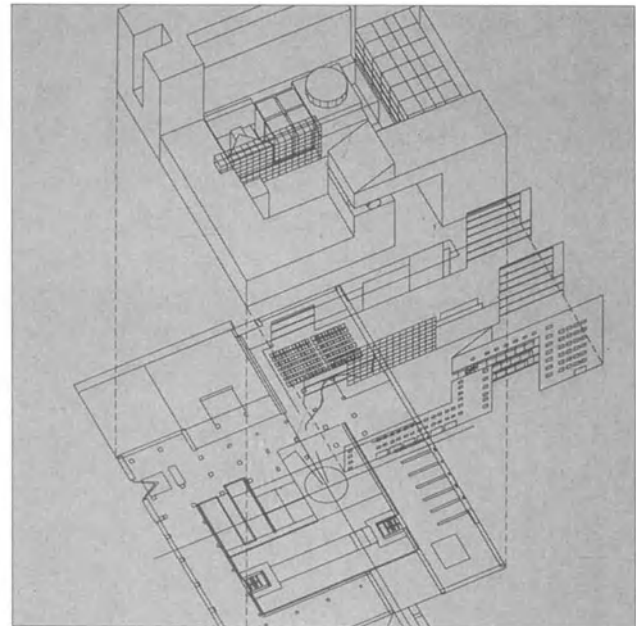
Projekt: Leandro Madrazo

Nach einem Text von Leandro Madrazo

Handskizze erster Ideen,
Eingangsbereich. Madrazo, 1992



Zerlegung des Entwurfs in ein System
von Elementen. Madrazo, 1992

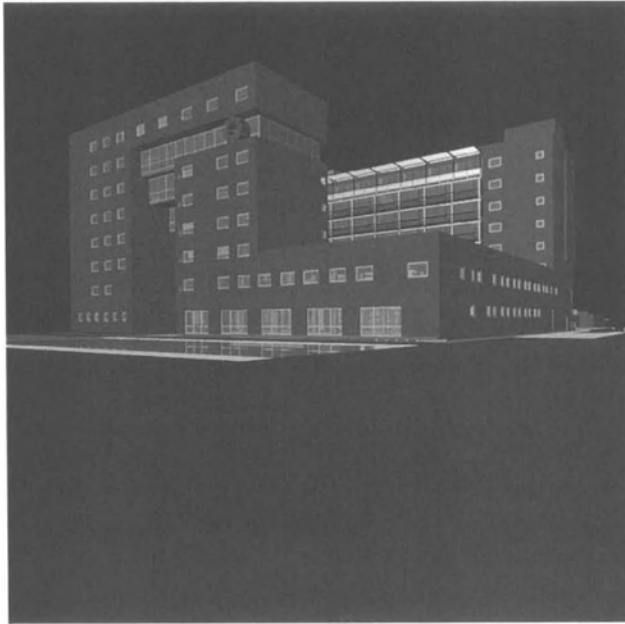


Dieses Projekt beschreibt einen Wettbewerb für eine Bank in Granada, Südspanien. Der Wettbewerb gab die Möglichkeit, einen Entwurf von Beginn an mit dem Computer zu entwickeln und unter praxisähnlichen Bedingungen zu arbeiten. Während des Entwurfs kam eine Reihe verschiedener Computerprogramme zum Einsatz. In der frühen Phase wurden zweidimensionale Zeichenprogramme verwendet, in der Entwurfsphase kamen dreidimensionale Modellierprogramme zur Anwendung, und die letzten Bilder wurden mit Rendering-Programmen für die photorealistische Darstellung erzeugt.

Das Grundstück für das neue Gebäude befindet sich am Rand Granadas in einem Neubaugebiet. Das Programm für das neue Gebäude war komplex: verschiedene Abteilungen mit eigenen Anforderungen in bezug auf Zugang, Flächen und Verbindungen untereinander. Die Gesamtfläche des Projekts betrug

30'000 m² bei einer maximalen Höhe von acht Geschossen. Die Wettbewerbsbeschreibung verlangte ein für die Stadt repräsentatives Gebäude. Aus diesem Grund machte der Entwurf von Beginn an eine klare Referenz an die Alhambra, die maurische Festung, als eines der Symbole von Granada. Wie es für die arabischen Architektur charakteristisch ist, gibt es in der Alhambra eine eindeutige Unterscheidung zwischen Außen- und Innenraum. Von außen besteht sie aus massiven Türmen und großen Toren. Von innen gesehen ist die Alhambra ein luxuriöser Palast, dessen Innenhöfe mit reflektierenden Wasserbecken und dessen Wände mit reich dekorierten Fliesen ausgestattet sind. Der extreme Gegensatz zwischen dem abweisenden Äußeren und dem einladenden Inneren wurde zu einem der bestimmenden Faktoren des Entwurfs. Die Skizzen zeigen die ersten Ideen. Das Schema basiert auf einem quadratischen Plan mit zentra-

Blick in das Innere durch die offene Ecke.
Madrazo, 1992, Rendering van der Mark, 1992



Der Eingang als große Öffnung in der Wand einer Festung.
Madrazo, 1992, Rendering van der Mark, 1992



lem Innenhof, der die islamische Architektur charakterisiert. Am Rand ist die Hauptmasse des Gebäudes konzentriert, was, wie bei der Alhambra, an den Festungscharakter erinnert. Der zentrale Innenhof, obwohl nicht vollkommen von ihnen umschlossen, wird durch diese Massen definiert. Eine Ecke der Mauer ist ausgeschnitten, um den Lichteinfall zu ermöglichen und die visuelle Verbindung zwischen innen und außen herzustellen.

Nach den Skizzen entstanden mit einem Volumenmodellierer dreidimensionale Modelle. Die Idee des aus einer Masse herausgeschnittenen Innenhofes läßt sich mit einem solchen Werkzeug gut realisieren. Die ursprüngliche Entscheidung für die auf einem quadratischen Plan und dem Innenhof basierende Typologie erwies sich dabei für die Arbeit am Computer als nützlich. Innerhalb der Grenzen des einmal etablierten Typs entstanden nun verschiedene Variationen.

Das Arbeiten mit dem Computer verlangte auch die Zerlegung des gesamten Objekts in seine grundlegenden Komponenten. Die Axonometrien zeigen dieses Ausgangsvokabular: die massiven Blockränder außen und die leichten Pavillons im Innern. Wie die kleinen Gebäude im 'Patio de los Leones' der Alhambra dienen diese Pavillons als Vermittler zwischen dem menschlichen Maßstab und dem Maßstab der Außenwände.

Parallel zur Entwicklung des dreidimensionalen Modells ging die Arbeit an abstrakten Kompositionen mit einem zweidimensionalen Zeichenprogramm weiter. Diese ursprünglichen Kompositionen spielten eine wichtige Rolle im Entwurfsprozeß. Sie trugen eine symbolische Bedeutung, die im Laufe des Entwerfens immer spezifischere Formen annahm. Die Pavillons waren beispielsweise anfänglich lediglich durch Platzhalter wie Quadrat oder Kreis angedeutet.

Weg zum Eingangspavillon im Patio. Madrazo,
1992, Rendering van der Mark, 1992



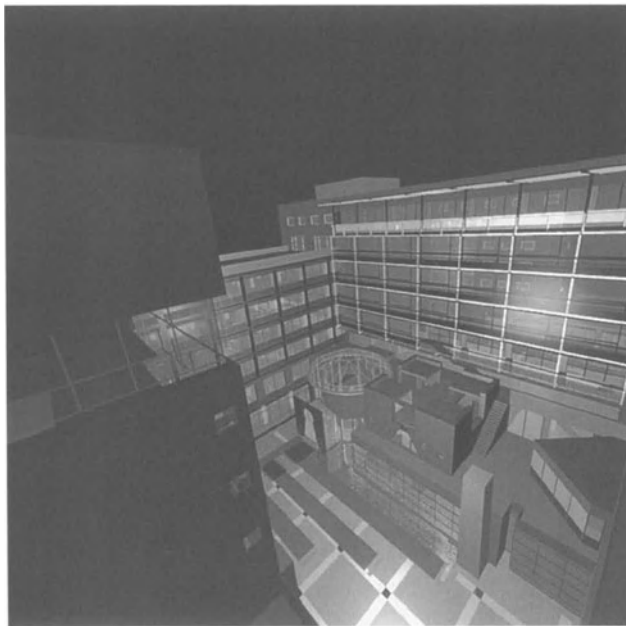
Blick nach oben im Eingangspavillon. Madrazo,
1992, Rendering van der Mark, 1992



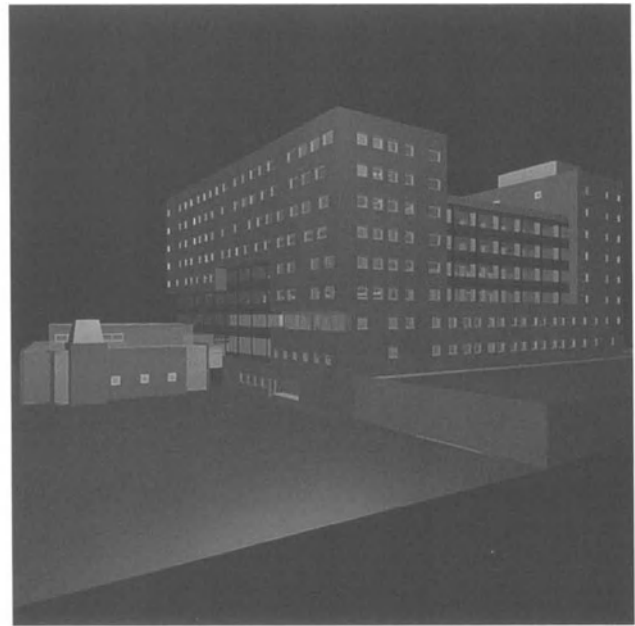
Erst später erfolgte ihre Transformation in architektonische Objekte. Diese Transformation von Geometrie in Architektur stellte sich als einer der wichtigsten Aspekte bei der Benutzung von Computern im Entwurf heraus. An einem ganz bestimmten Punkt im Entwurfsprozeß, wenn Maßstab und konstruktive Überlegungen ins Spiel kommen, nehmen die geometrischen Formen erste architektonische Bedeutung an. So verwandelte sich der Zylinder im ursprünglichen dreidimensionalen Modell in den Eingangspavillon. Das Dach dieses Pavillons besteht aus einem Raumtragwerk, das nach dem Stern im Logo der Bank geformt ist. Die Tür des Pavillons hat die Form eines Schlüssellocks, eine Referenz an den Schlüssel zum Safe der Bank, besonders aber an die arabische Form des Bogens.

Das Erkennen der inneren Struktur des Entwurfs ist eine Voraussetzung für den erfolgreichen Einsatz des Computers [Madrazo 1993b]. Diese Struktur kann und soll im Computer direkt repräsentiert sein. Dafür ist das hierarchische Modellieren der Komponenten wichtig, wodurch aus einfachen Elementen komplexere erzeugt werden können. Die Fassade des Innenhofs ist ein besonders gutes Beispiel. Mit dem Instrument der Typen und Variationen entstand aus Trägern, Stützen und Geländern ein zusammengesetztes Objekt, das die Fassade bildet. Durch Variation der Einzelteile wurde die endgültige Form nach vielen dreidimensionalen Variationen gefunden. Obwohl bei der Entwicklung dieses Entwurfs das dreidimensionale Modell im Vordergrund stand, kamen auch die traditionellen Darstellungen wie Plan,

Blick in den Patio und auf die Pavillons.
Madrazo, 1992, Rendering van der Mark, 1992



Hauptgebäude mit angelagertem Auditorium.
Madrazo, 1992, Rendering van der Mark, 1992



Schnitt und Ansicht zur Anwendung. In der Tat war die gleichzeitige Verwendung dieser Projektionen bei der Komplexität des Projekts notwendig. Wegen der Verwendung eines allgemeinen, nicht für die Produktion von Architekturzeichnungen optimierten CAD-Programms trennten sich kurz vor der Abgabe die Pläne, Ansichten und Schnitte vom Modell und wurden unabhängig voneinander fertiggestellt. Die dadurch entstehende Inkonsistenz bei etwaigen Änderungen wurde in Kauf genommen.

Die photorealistischen Darstellungen, die in einem professionellen Rendering-Programm entstanden, spiegeln die Hauptideen des Entwurfs wider, insbesondere den Kontrast zwischen innen und außen. Das Äußere zeigt das Gebäude als einen massiven Backsteinbau, mit einem kleinen Hinweis auf das

Innere an der offenen Ecke. Nähert man sich dem Gebäude von dieser Richtung, dann wird das Innere und damit der Gegensatz zur abweisenden Fassade sichtbar. Der Eingang zeigt sich deutlich auch im Bodenbelag, der auf das Nadelöhr des eigentlichen Eingangs in das Gebäude hinführt. Im Innenhof sind Glas, Metall, Wasser und Keramikplatten bestimmend. Das Resultat ist ein von vielerlei Reflexionen erfüllter Raum.

Computer-Renderings können ein ausgezeichnetes Werkzeug zur Darstellung des Materials und der räumlichen Qualität sein. Sie zeigen allerdings auch gnadenlos die weniger erfolgreichen Aspekte des Entwurfs, falls die gewählten Proportionen oder der Detaillierungsgrad für diese Darstellungsart nicht angemessen sind [Madrazo 1993a].

Projekt: Simulation eines neuen Chemiegebäudes

Architekten: Mario Campi und Franco Pessina

Modell: Sharon Refvem

Rendering und Animation: Eric van der Mark

Der Höggerberg mit dem simulierten Modell des neuen Chemiegebäudes im Vordergrund links. Refvem und van der Mark, 1992

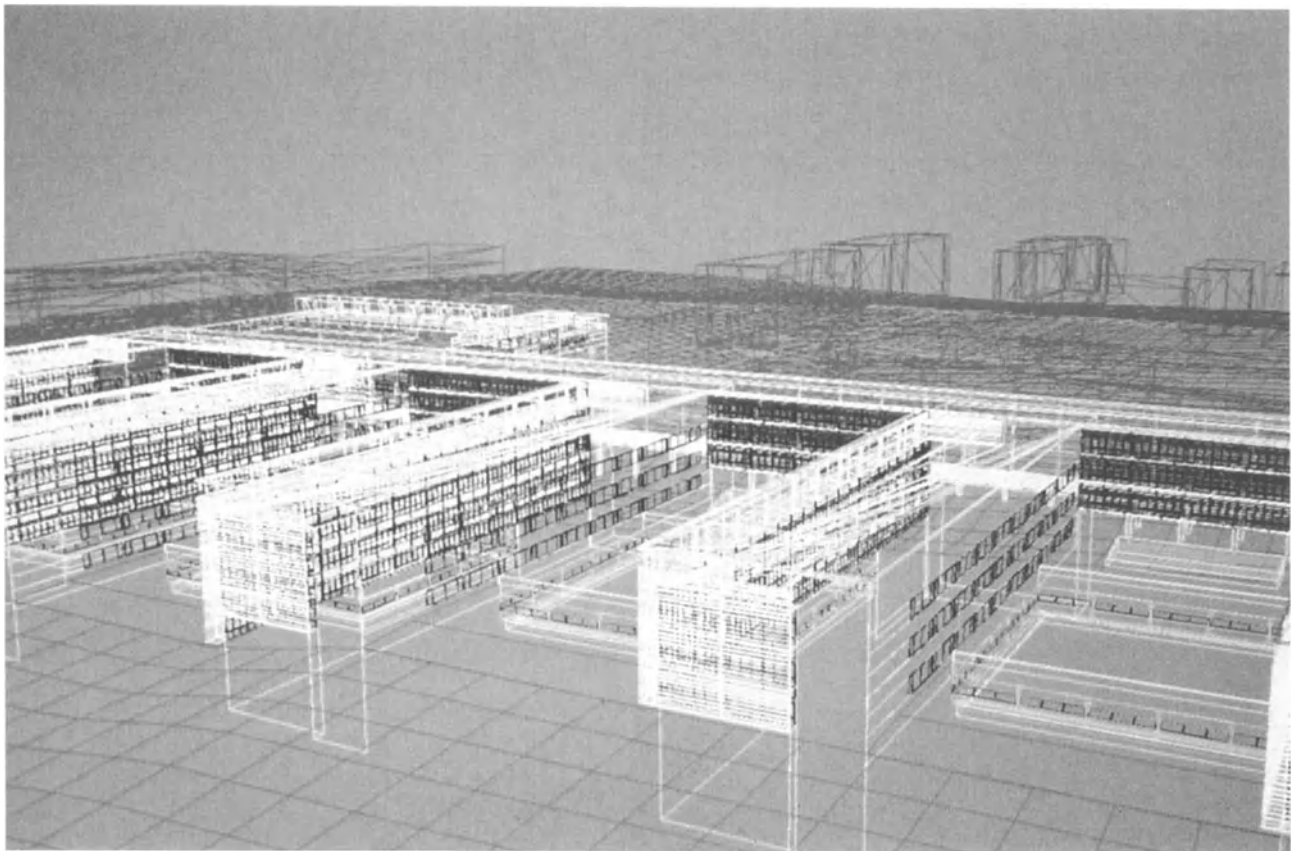
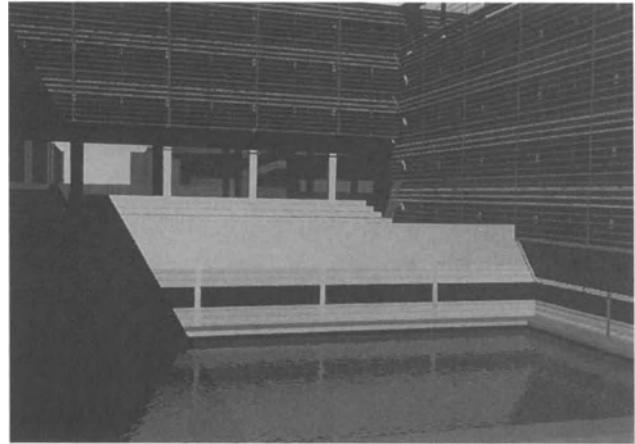


Das Projekt war ein Testfall in verschiedenen Beziehungen. Es demonstrierte, daß die von uns entwickelten Instrumente nicht nur für den Unterricht, sondern bei entsprechendem Wissen auch für den praktischen Entwurf von Nutzen sind. Es zeigte auch die Wichtigkeit der Verwendung verschiedener Programme und Techniken sowie die Bedeutung menschlicher Arbeitsteilung und Kommunikation.

Auf dem Höggerberg-Areal der ETH Zürich soll ein neues Chemiegebäude entstehen. Nach einem Wettbewerb kam das Projekt der Architekten Campi-Pessina zur weiteren Bearbeitung. Wegen der Größe des Projekts entstand seitens Quartiervereinen, Politikerinnen und Politikern sowie der ETH-Leitung das Verlangen nach einer Reihe von Simulationen, die verschiedenste Aspekte des neuen Gebäudes zeigen sollten. Diese Aufgabe bot die

Rendering eines
Innenhofs (oben),

Drahtmodell des neuen
Chemiegebäudes
(unten). Refvem und
van der Mark, 1992



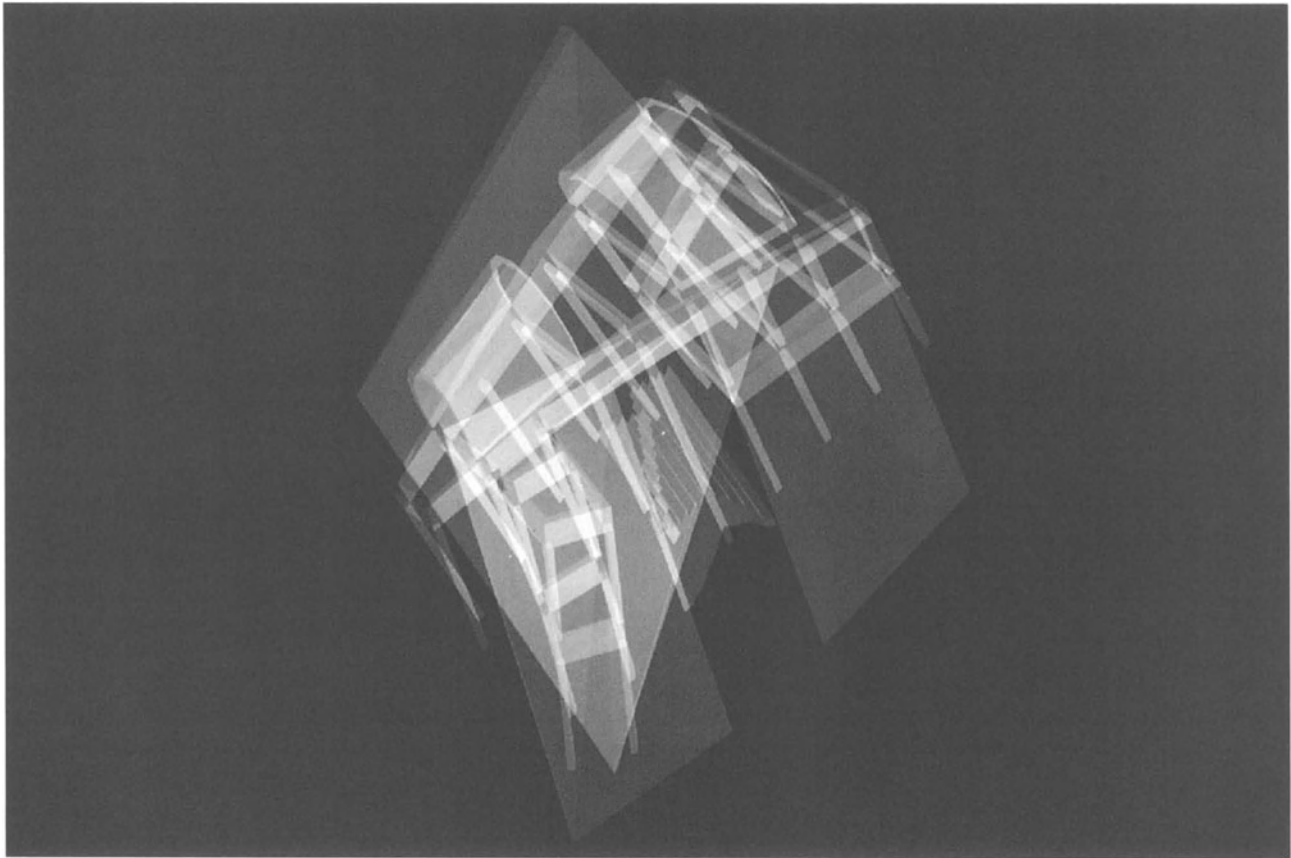
Gelegenheit, das Instrument der Typen und Variationen an einem praktischen Bauprojekt einzusetzen. Zunächst entstand so ein grobes Massenmodell aller bestehenden Gebäude auf dem Hönningerberg und der umgebenden Topographie. Danach modellierte Sharon Refvem die einzelnen Gebäudeteile als Verfeinerung der Typen. Durch selektive Erhöhung des Detaillierungsgrades konnten so verschiedene Aspekte des Projektes evaluiert

werden, wodurch auch die Architekten Erkenntnisse gewann. Das gesamte Projekt besteht als dreidimensionales Modell und kann in Echtzeit begutachtet werden, wenn Fragen zu bestimmten Aspekten entstehen. Änderungen in der Position und Größe der Gebäudeteile sind auf diese Weise schnell zu visualisieren. Die aus demselben Modell erzeugten Videos dienen der Öffentlichkeit als wichtige Informationsgrundlage.

Ausstellung: New Realities

Samples: Eric van der Mark, Zoran Sladoljev, Sharon Refvem, Florian Wenz, Antonino Saggio

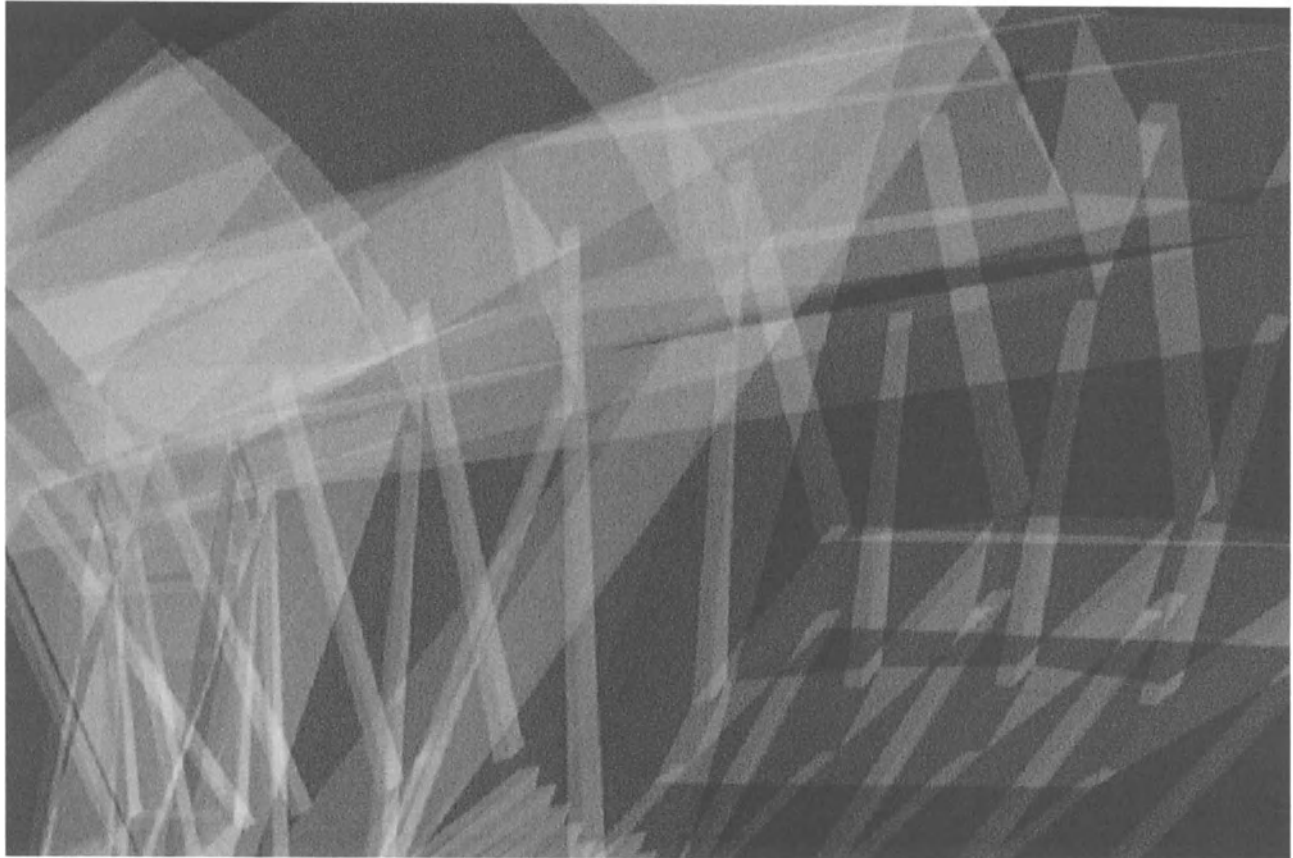
Nicht-lineare Transformation des Modells eines Gebäudes von
Tadao Ando. Modell Vezin, 1990, Animation van der Mark, 1993



Neue Wirklichkeiten sind am besten in den Arbeiten mit intelligenten Programmen am graphischen Computer sichtbar. 1992 und 1993 fand an der Schule für Gestaltung in Zürich die Ausstellung 'New Realities' statt. War 1992 das Thema ein allgemeiner Einblick in die neuen Techniken, so stand 1993 ganz im Zeichen des CAAD. Für beide Ausstellungen wählten wir eine Reihe von Animationen aus. Die Video-Samples entstanden aus den verschiedensten Anlässen. Allen gemeinsam ist die Absicht, etwas noch nicht oder nicht mehr Bestehendes in anderer Sicht darzustellen und bei bestehenden Bauten Aspekte zum Ausdruck

zu bringen, die in traditionellen Darstellungsweisen verborgen bleiben. 'Ando' entstand aufgrund einer Idee am Ende einer Seminarwoche 'Beyond Reality' im Dezember 1991. Ich erinnerte mich an die Möglichkeit nicht-linearer Transformationen in einem High End-Visualisierungsprogramm, das wir bisher nur für Renderings eingesetzt hatten. Zugleich hatten wir ein dreidimensionales Computermodell des Gebäudes von Tadao Ando, das meiner Meinung nach nicht die dynamischen Aspekte des Gebäudes zeigte. Die Umwandlung des Modells in ein neutrales Format, die anschließende Übertragung in den

Ausschnitt aus dem nicht-linear transformierten Modell.
Modell Vezin, 1990, Animation van der Mark, 1993



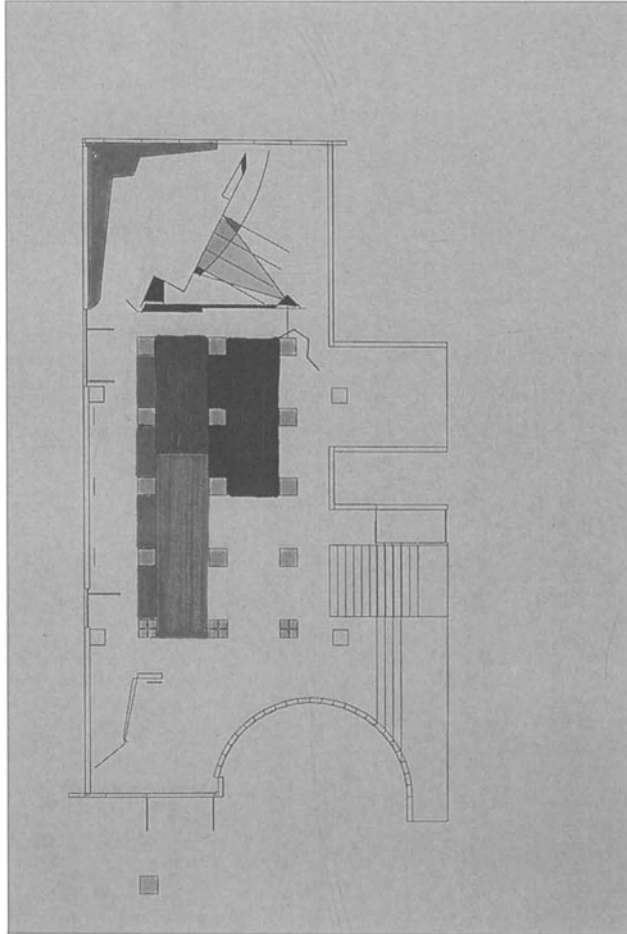
Modellierer und die direkt im Anschluß daran ausgeführte nicht-lineare Transformation erfolgte innerhalb weniger Minuten, hatte aber auf uns alle eine große Wirkung, da aus der Kombination verschiedenen Wissens und verteilter Fähigkeiten etwas Neues entstanden war, das uns alle faszinierte. 'Terragni' ist eine Arbeit von Antonino Saggio. Er konzipierte das Sample als Video. Es entstand auf einem Macintosh und integriert gescannte Images und dreidimensionale Modelle. Die didaktischen Aspekte des Samples sind unverkennbar; es gehört also in den Bereich architektonischer Teachware. Alle Beispiele wurden

von Eric van der Mark vertont. Mein Verhältnis zu dieser Art der Darstellung ist nach wie vor ambivalent. Ich vertrete den Standpunkt, durch höhere Computerleistung das interaktive Erleben der Objekte zu ermöglichen, wozu auch die Sichtbarmachung nicht-geometrischer und normalerweise unsichtbarer Phänomene gehört. Mir ist unwohl bei dem Gedanken, die Vielzahl der Möglichkeiten und Qualitäten eines Computermodells in ein Video von wenigen Minuten zu bannen. Auf lange Sicht wird sich die interaktive Betrachtung für den neugierigen Besucher durchsetzen.

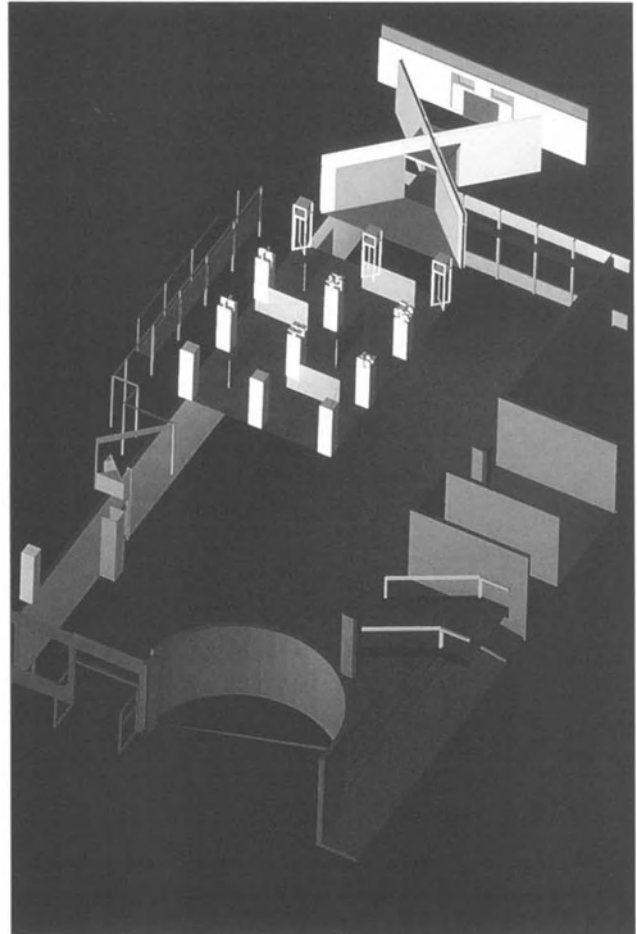
Ausstellung: Lineamenta - CAAD

Projekt: Werner Oechslin und Gerhard Schmitt
Ausstellungskonzept und Text: Leandro Marazo

Erste Idee: Verschiedene
Objektsysteme. Madrazo, 1991



Ausarbeitung: Ein System, bestehend aus Untersystemen.
Madrazo, 1991, Rendering van der Mark, 1991

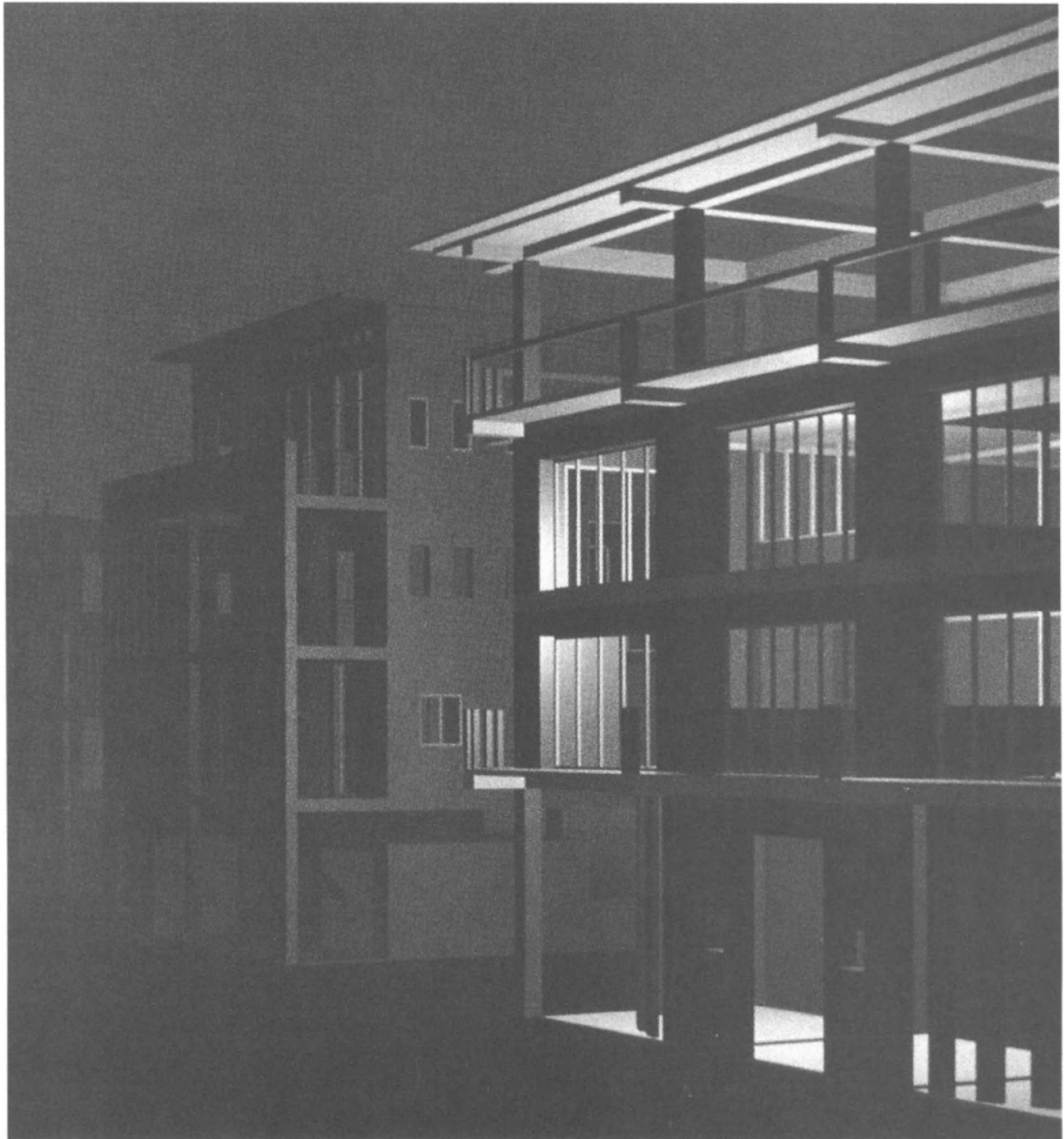


Im Sommer 1991 fand an der ETH Zürich die Konferenz CAAD futures '91 statt [Schmitt 1992b]. Die Ausstellung Lineamenta-CAAD war Teil dieser Konferenz und wurde gemeinsam vom Institut für Geschichte und Theorie der Architektur (gta) und von der Professur für Architektur und CAAD veranstaltet. Die Ausstellung hatte zwei Ziele: zum einen die Präsentation der seit der Etablierung der Professur für Architektur und CAAD entstandenen Arbeiten, die sich auf die Anwendung des Computers in der Architektur bezie-

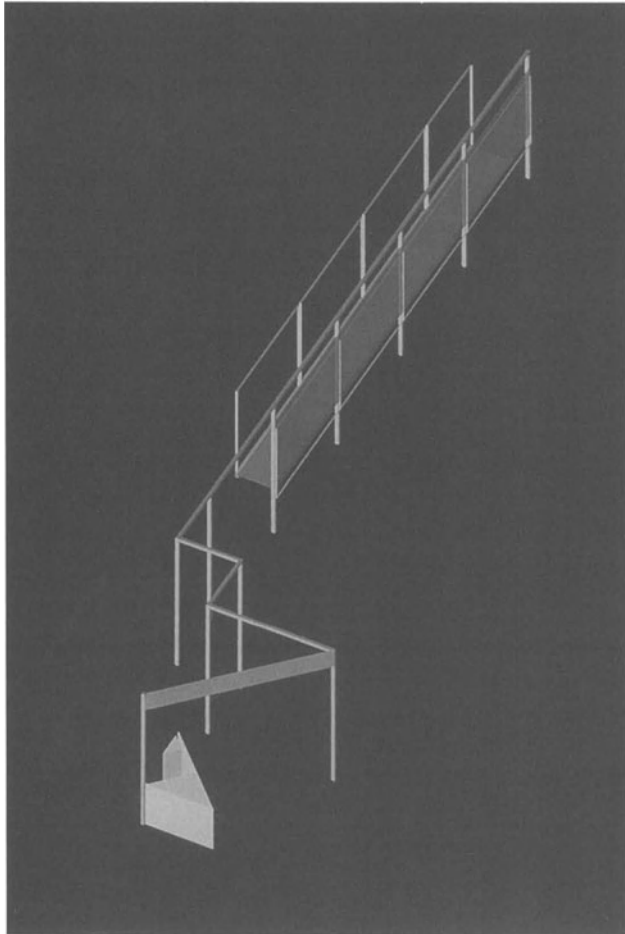
hen; zum anderen den Versuch, eine Verbindung zwischen dem modernen Gebrauch von Computern in der Architektur und der in der Architekturtheorie verankerten Idee von Architektur als Wissenschaft zu schaffen.

Der dem Gebiet CAAD gewidmete Teil der Ausstellung gliederte sich in die Abschnitte Forschung, Lehre und Praxis. Die Arbeiten wurden durch Poster, Farbvergrößerungen, Dias und Computeranimationen präsentiert.

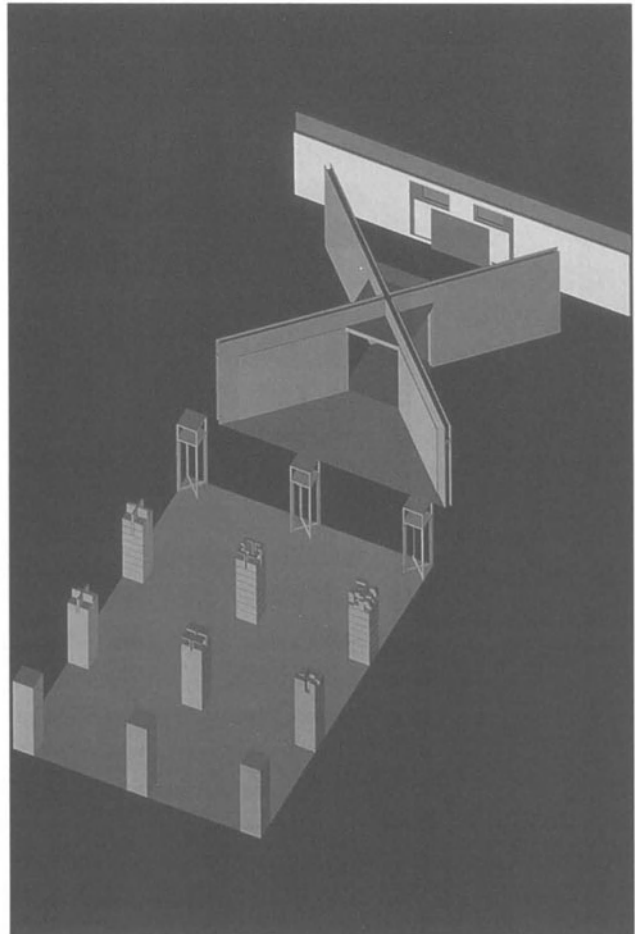
Titelblatt der Ausstellung Lineamenta - CAAD. Studentinnen und Studenten im Sommersemester 1990. Rendering van der Mark, 1991



Das Untersystem Portiko. Madrazo,
1991, Rendering van der Mark, 1991



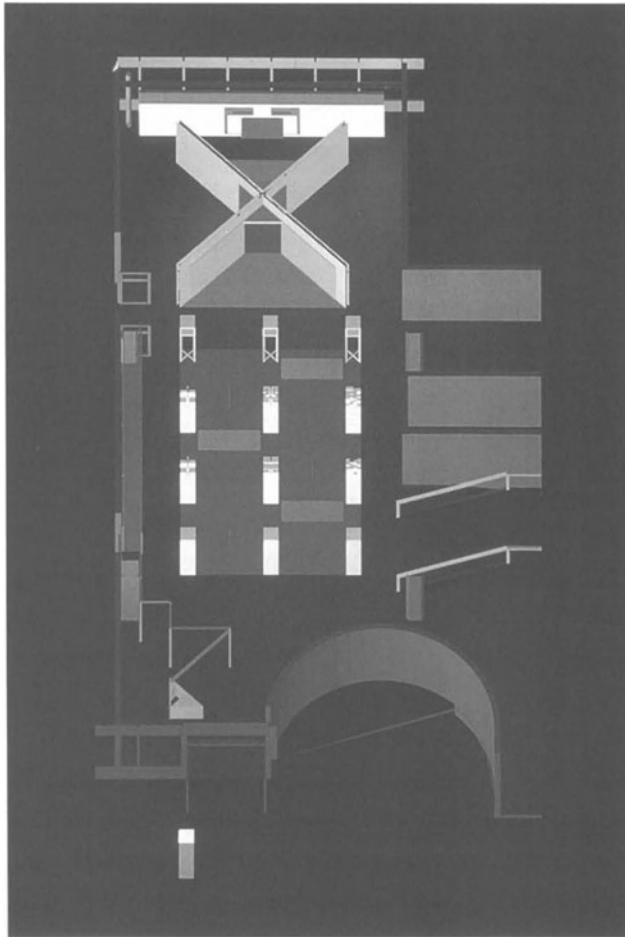
Interaktion zwischen den Untersystemen Stütze und
Kreuz. Madrazo, 1991, Rendering van der Mark, 1991



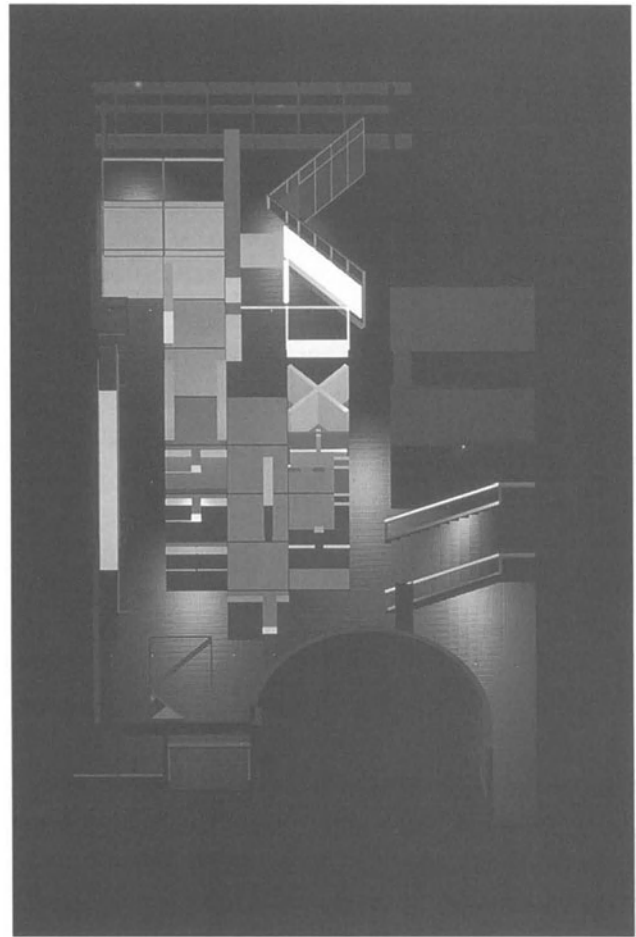
Der Entwurf der Ausstellung bot zum erstenmal die Chance, die an der Professur für Architektur und CAAD entwickelten Methoden und Instrumente an einem praktischen Projekt zu testen. Der Entwurf besteht aus Objekten, die auf einer Ausstellungsebene platziert sind. Diese sind nach vier verschiedenen Systemen organisiert: ein Säulenraster, ein Portiko, ein Kreuz und ein Set transparenter Paneele. Diese Systeme sind Teil eines hierarchisch strukturierten, abstrakten Organismus. Die gesamte Ausstellung ist

ein Objekt, das seinerseits aus verschiedenen zusammengesetzten Objekten besteht. Jedes System von Elementen hat innerhalb der Ausstellung eine spezifische Funktion. Die Paneele und Zeichnungen wurden entlang des Portikos platziert. Ein verdunkelter Raum für die Projektion der Dias entstand innerhalb eines der vier Segmente des großen Kreuzes. Graphiken und Diagramme, die sich auf CAAD beziehen, wurden auf die von der Decke hängenden Paneele platziert.

Axonometrie des ersten Entwurfsvorschlags.
Madrazo, 1991, Rendering van der Mark, 1991



Axonometrie des letzten Entwurfsvorschlags.
Madrazo, 1991, Rendering van der Mark, 1991

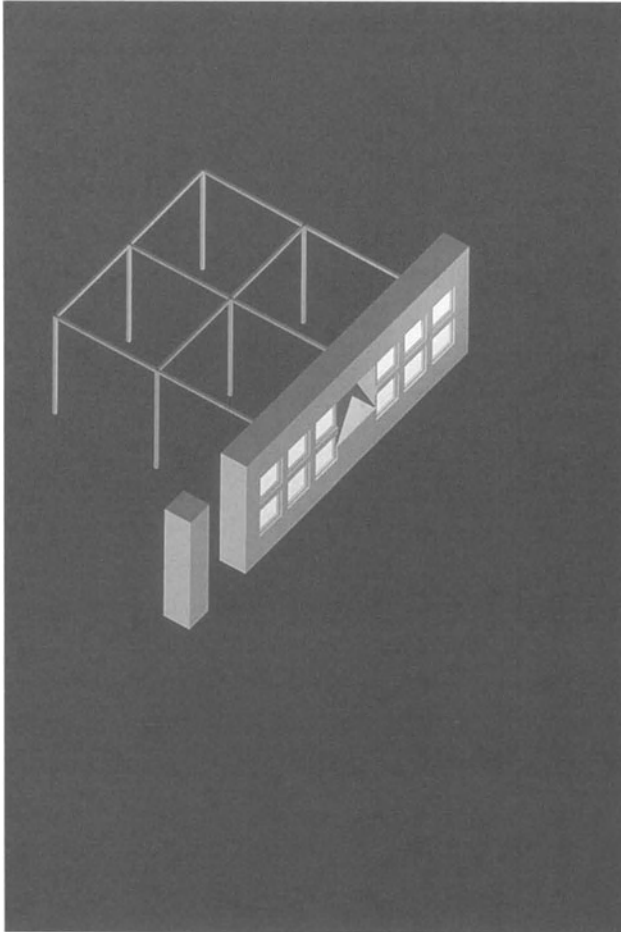


Jedes System in dem abstrakten Organismus hatte seine eigene Autonomie und seine eigenen Entstehungsregeln [Madrazo 1993b]. Im Säulennaster etwa bestand die erste Reihe aus einfachen aufeinandergesetzten Kuboiden. In der zweiten Reihe erhielten die Säulen die klassische Aufteilung von Basis, Schaft und Kapitell. In der dritten Reihe bewirkten die verfeinerte Ausbildung der Verbindungen und die komplexen Kapitelle eine neue Interpretation des Themas Säule. Dieser Fortschritt im Detaillierungsgrad der Säulen

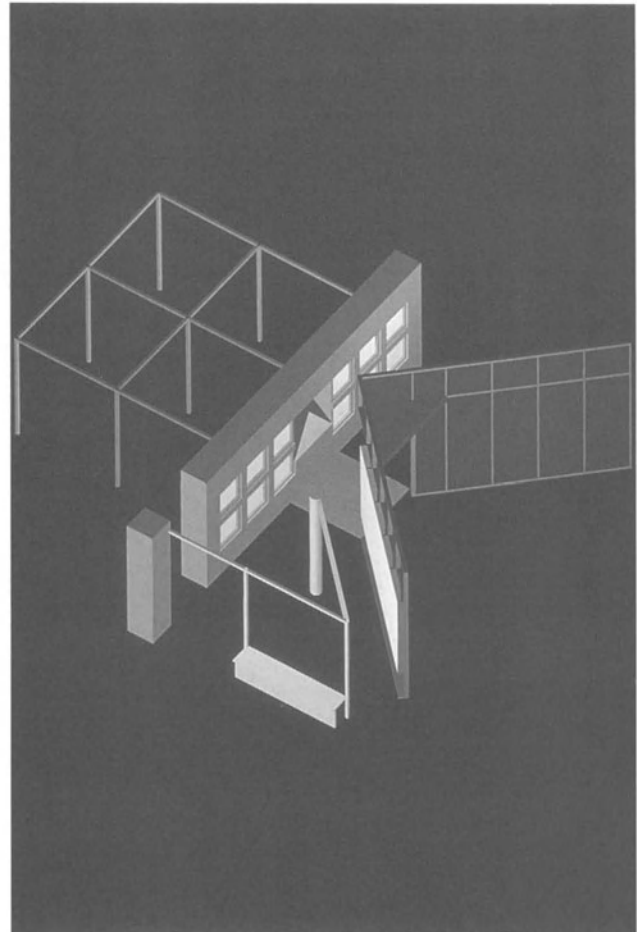
war die Realisierung der im Diplomwahlfach CAAD Principia gelehrteten Idee. Obwohl die Systeme voneinander unabhängig waren, interagierten sie. Als Resultat einer Interaktion mit dem naheliegenden Kreuzsystem beendeten die Säulen in der letzten Reihe ihren Fortschritt im Detaillierungsgrad und nahmen stattdessen die Kreuzform des nächsten Systems an.

Der als ein abstrakter Organismus geplante Entwurf unterhielt eine dialektische Beziehung mit seinem

Hierarchie von Elementen: Wand und Raster.
Madrazo, 1991, Rendering van der Mark, 1991



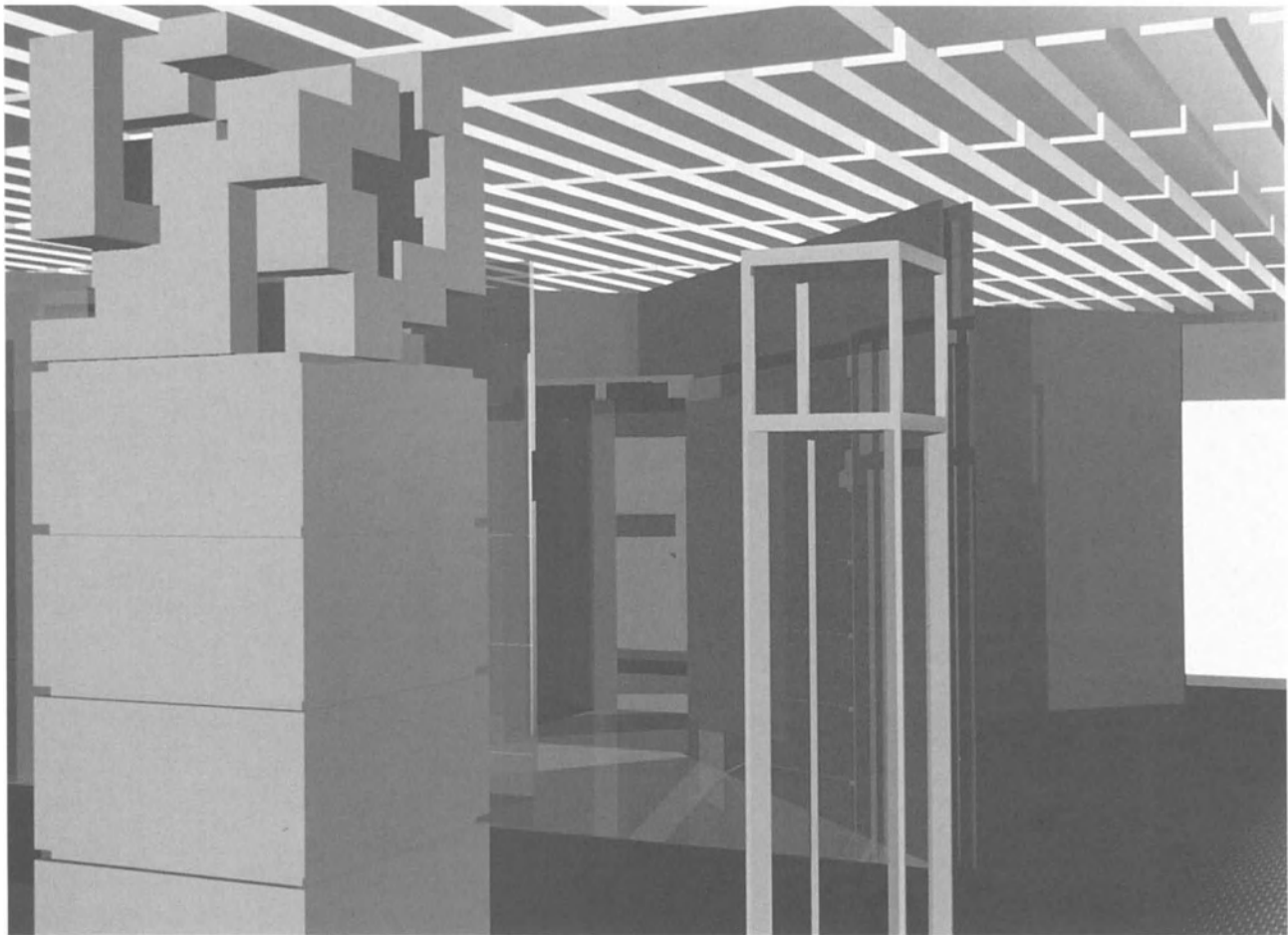
Hierarchie von Elementen: Gruppierung von Objekten um die Wand. Madrazo, 1991, Rendering van der Mark, 1991



Kontext. Das Portiko-System verlief parallel zur Längsseite des Ausstellungsraumes. Sobald der Portiko sich dem Eingang der Lobby näherte, entfernte er sich von der Wand, so, als werde er vom Eingang angezogen. Ein Gleichgewicht wurde in dem Moment erreicht, in dem der Portiko in einem Winkel von 45 Grad auf den Eingang zulief. In diesem Fall war der Eingang das Element des Kontexts, mit dem das Portiko-System die dialektische Beziehung herstellte.

Einige Systeme innerhalb des abstrakten Organismus reagierten direkt auf ausgewählte Eigenschaften des Kontexts, ignorierten aber andere. Der Portiko zum Beispiel reagierte auf den Einfluß des Eingangs, vernachlässigte aber den Einfluß des Kreuz-Systems. Wir bezeichneten diese Eigenschaft als Kodifizierung der Elemente und Systeme. In der Repräsentation der Objekte im Computer in einer objektorientierten Weise waren diese Kodifizierungen Teil der Objekte selbst. So enthielt jedes Objekt des Entwurfs die Kodi-

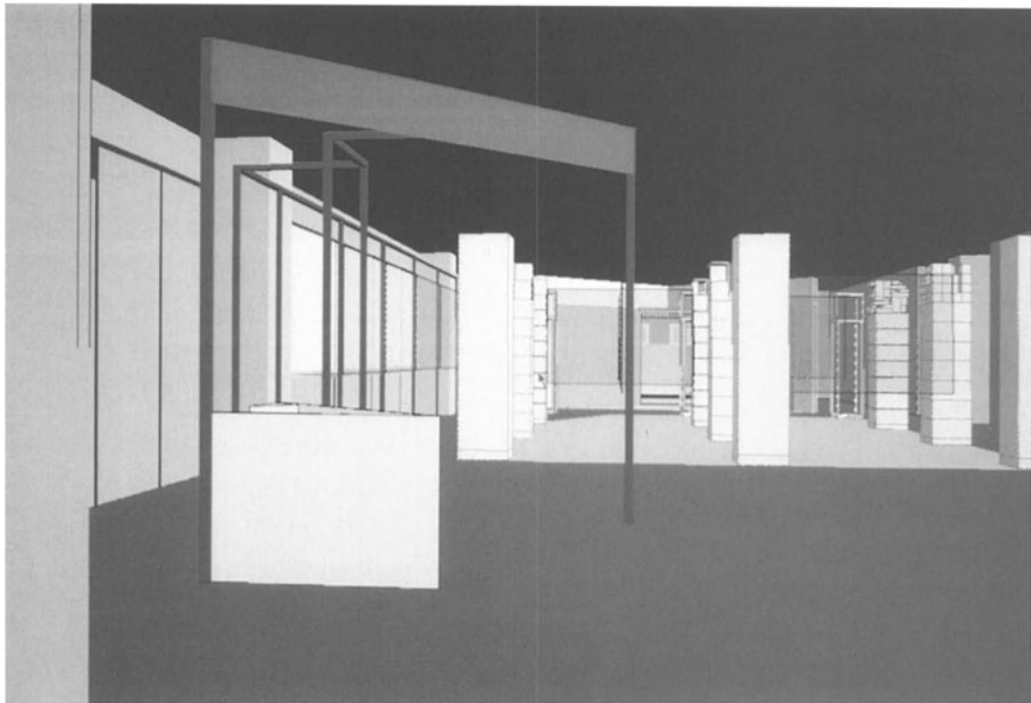
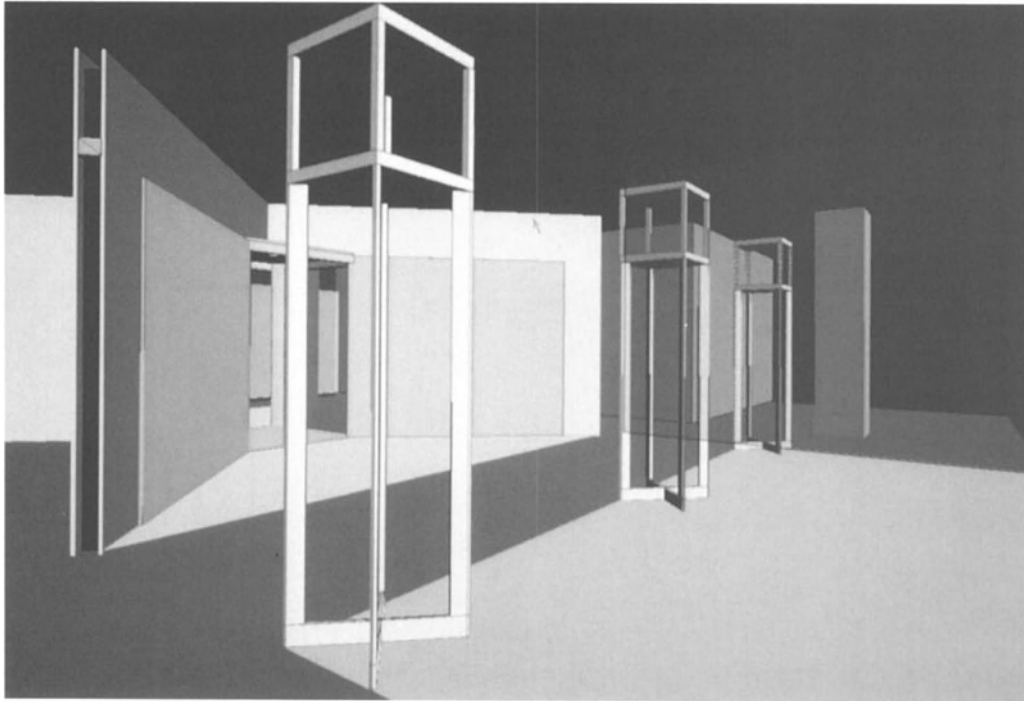
Detail einer Säule, generiert mit einer Froebel-Grammatik.
Madrazo, 1991, Rendering van der Mark, 1991



fizierung, welche die Beziehung des Objektes mit seiner Umgebung bestimmte [Madrazo 1992b]. Aufgrund von Änderungen im Programm, sowie von zeitlichen und finanziellen Beschränkungen, mußte ein zweiter Vorschlag für die Ausstellung gemacht werden. Trotz formeller Änderungen behielt dieser zweite Vorschlag die Essenz der ersten Entwurfsidee bei. Obwohl einige der Teile durch neue ersetzt wurden, blieb die Struktur des abstrakten Organismus in verschiedenen Teilen die gleiche. Die Säulen bei-

spielsweise wichen einfachsten, bereits vorhandenen Elementen, die auf einem Raster von Stahlplatten angeordnet werden sollten. Diese Objekte trugen die Paneele mit den Arbeiten der Studentinnen und Studenten. Das ursprüngliche Kreuz wurde durch ein komplexeres System von um eine massive Wand platzierten Elementen ersetzt. Ebenso blieb die Diaprojektion in einem Sektor des Kreuzes. Auch der Portiko blieb wie im ersten Projekt erhalten. Lineamenta - CAAD wurde zur Wanderausstellung.

Übergang zwischen dem Untersystemen Kreuz und Säule. Madrazo, 1991
Blick vom Eingang: Der Portico empfängt den Besucher. Madrazo, 1991



Photorealistisches Rendering eines Schemas: Säulen und von der Decke gehängte transparente Paneele. Madrazo, van der Mark, 1991



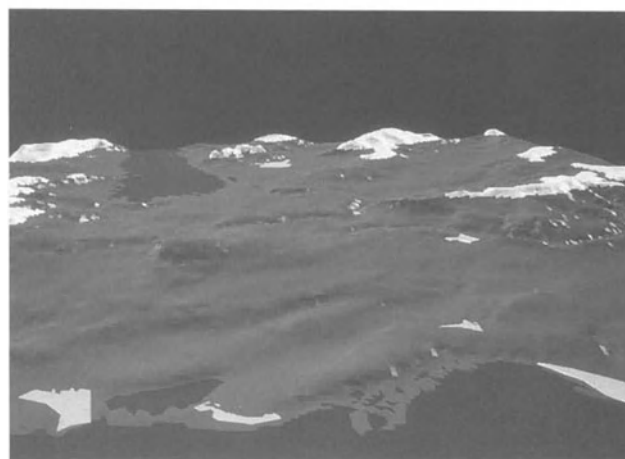
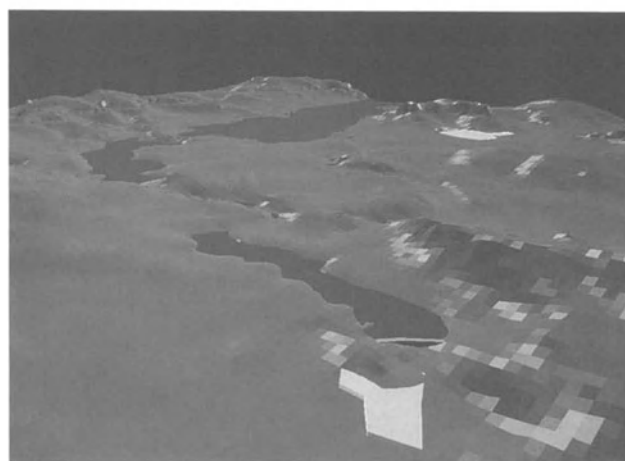
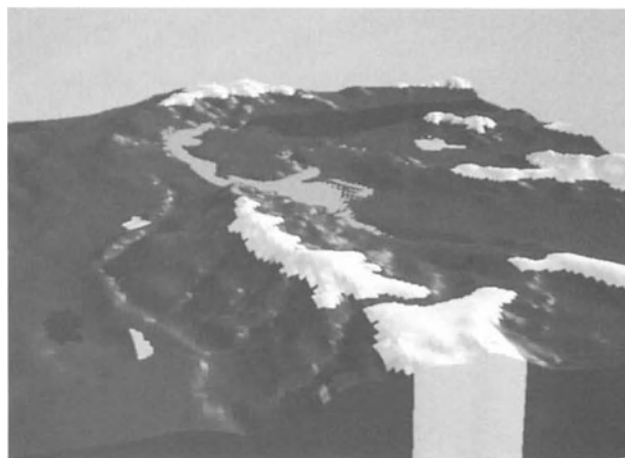
Projekt: Der Traum vom großen blauen Wasser

Projekt: Karl Saurer

Bearbeitung: Eric van der Mark

Ausgehend von einer Idee des Einsiedler Regisseurs Karl Saurer entstand die Aufgabe, mögliche Konsequenzen der Schaffung eines künstlichen Sees zu visualisieren. Ausgangspunkt waren verschiedene Vorschläge für einen Stausee in der Zentralschweiz zu Beginn dieses Jahrhunderts. Das schließlich ausgeführte Projekt ließ den heute als Sihlsee bekannten Stausee bei Einsiedeln entstehen. Die Planungen lagen in unterschiedlichem Detaillierungsgrad vor, vom ausgearbeiteten Projekt bis zur schematischen Skizze. Wir setzten uns zum Ziel, alle Projekte digital darzustellen, um so visuelle Vergleichsmöglichkeiten zu haben.

Der erste Schritt war die Erarbeitung eines dreidimensionalen Computermodells der Landschaft um den heutigen Sihlsee bis hinunter zum Zürichsee. Danach 'stauten' wir für die einzelnen Vorschläge die Sihl so lange, bis der Wasserspiegel die vorgesehene Höhe erreichte. Das Wasser breitete sich damit korrekt im Computermodell aus und zeigte die Verschneidung mit der bestehenden Landschaft an. So gelang die genaue Definition der Uferlinie. Diese Linie übertrugen wir in digitale Dia-Aufnahmen, die bei einem Flug mit dem Hubschrauber entstanden waren. Durch Bildverarbeitungsmethoden stellten wir den ursprünglichen Zustand im Tal der Sihl wieder her und füllten die neuen Stauseen mit Wasser. Auf diese Art erstand das alte Sihltal mit Feldern und Höfen wieder, während der kleine Ort Egg im Wasser versank. Ein zweiter Vorschlag sah bei dem Dorf Schindellegi eine gewaltige Staumauer von 80 Metern Höhe und über 1200 Meter Länge vor. Das Endresultat waren photorealistische Einzelbilder und eine Animation. Die Animation zeigte lediglich das dreidimensionale Modell, aus dem die Uferlinien gewonnen wurden. Es ließ aber auch Fehleinschätzungen bei den Alternativen über den Verlauf der Uferlinie erkennen.



Modell mit Sihlsee im Hintergrund,
Vorschlag Staumauer bei Egg,
Blick vom Zürichsee auf den Sihlsee. Van der Mark, 1991

Blick auf den heutigen Sihlsee von Norden.
Simulierter Stausee über Egg, wiederhergestellte Landschaft unter dem heutigen Sihlsee. Van der Mark, 1991

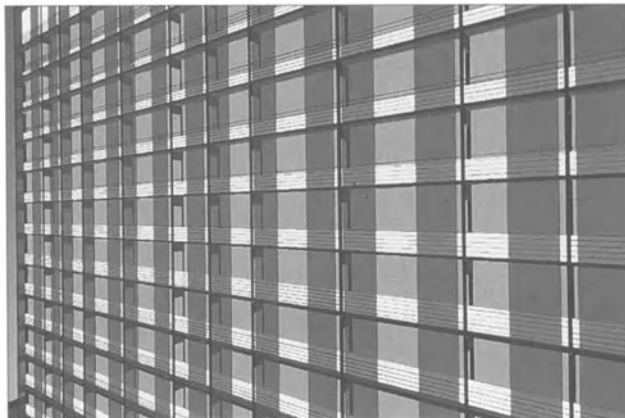
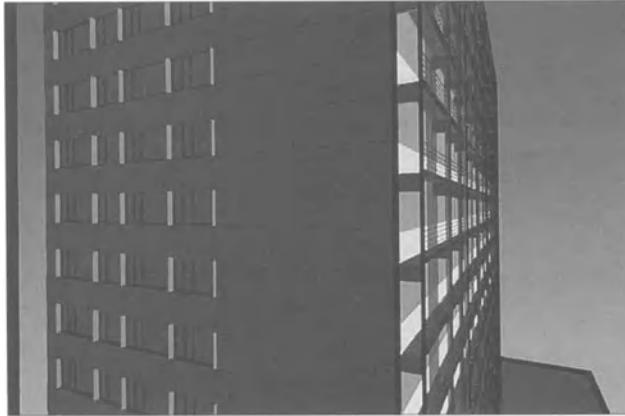


Projekt: Triemli

Architekten: Sumi und Burkhalter

Modell: Sharon Refvem

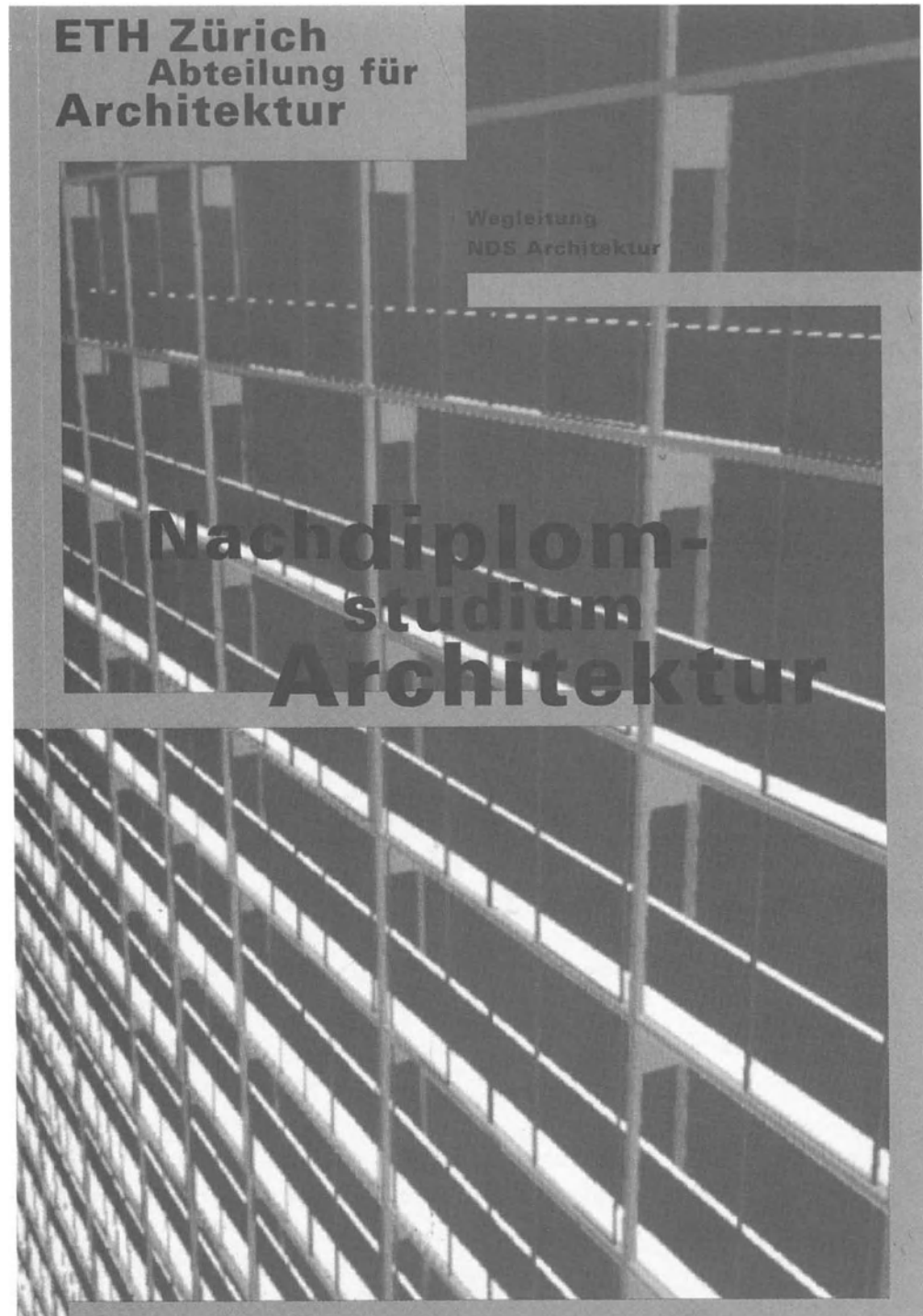
Animation: Eric van der Mark



Das Projekt 'Triemli' entstand als praktische Anwendung in Zusammenarbeit mit dem Architekturbüro Sumi und Burkhalter in Zürich. Das Büro hatte den Auftrag, für das Zürcher Triemli-Spital eine neue Fassade zu entwickeln. Nicht nur die äußere Hülle und die Farbgebung bedurften der Modifikation, sondern auch das damit verbundene Energiekonzept. Die Forschungskomponente liegt in der Erprobung und Verfeinerung der von uns entwickelten objektorientierten Variations- und Substitutionsmethoden an einem realen großen Bauwerk. In vielen Sitzungen mit den Architekten entstand das vorliegende, farblich abgestimmte Computermodell, von dem

während des Prozesses hauptsächlich orthogonale Projektionen entstanden. Das im Video gezeigte dreidimensionale Computermodell war die Basis für alle Schnitte und Ansichten. Die Farben der neuen Bauteile definierten die Architekten und Sharon Refvem gemeinsam interaktiv am Bildschirm und am Video-Monitor. Die Animation ist eine Echtzeit-Aufzeichnung der Abläufe, mit denen sich die Beobachter am Computer durch und um das Triemli-Spital bewegen können. Die simulierte Fassade fand später als Titelblatt für das Nachdiplomstudium der Architekturabteilung Verwendung - ein interessantes Beispiel für die Umnutzung einer Simulation.

Bilder aus einem Video
über den projektierten
Umbau des Triemli-Spitals.
Modell Refvem, 1992,
Animation van der Mark,
1993



Titelblatt der
Informationsbroschüre für
das Nachdiplomstudium
Architektur der ETH Zürich
mit dem Motiv der Triemli-
Fassade. Frei, 1992

Projekt: Rote Drachenstele

Projekt und Text von Urs B. Roth

Eine Ausstellung der Gesellschaft Schweizerischer Maler, Bildhauer und Architekten G.S.M.B.A. im Zürcher Kunsthaus gab den 1992 aufgenommenen Künstlern die Gelegenheit, sich je mit einem Werk der Öffentlichkeit vorzustellen.

Keine leichte Aufgabe; nicht nur wegen der kunterbunten Vielfalt der gestalterischen Ansätze, die da offen aufeinanderzuprallen drohten, sondern auch wegen der beschränkten räumlichen Verhältnisse. Mit einer Figur, die sich hauptsächlich vertikal entwickelt, konnte ich der optischen Enge eventuell entkommen.

Ich träume: ich sehe einen Turm, hölzern, schief im Raum stehend, der Schwerkraft spottend, rot. Gerade als konkreter Künstler, dessen tägliches Brot die Ratio ist, schätze und akzeptiere ich gerne meine Traumbilder. Sie sind suggestive Richtschnur für meine Arbeit, Ausdruck einer irrationalen Welt, zu deren Quellen ich keinen direkten Zugang habe [Baumeister 1947]. Ich greife zurück auf eine Figur, die ich 1988 entwickelt hatte: die 'Drachenstele' [Schaad 1988]. Sie entsteht aus der regelmäßigen, alternierenden Anordnung von speziellen rechts- und linkshändigen isodrischen Tetraedern. Eine Helixanordnung von kongruenten, symmetrischen Drachenvierecken bildet ihre Oberfläche. Diese Oberfläche ist topologisch identisch mit der Parkettierung der Ebene durch kongruente Viereck und ich konnte zeigen, daß sich die räumliche Figur aus der Ebene (zum Beispiel einem Stück Papier) durch gekoppelte vierachsige Faltungen generieren läßt.

Soweit ist die Drachenstele erst ein geometrisches Prinzip, aber noch keine Plastik. Entstanden ist sie mit konventionellen Mitteln: Skizze, erste Berechnungen, Aufrisse an der Zeichenmaschine, unzählige Kartonmodelle. Und als wichtigstes Hilfsmittel: ein fast schon historisch zu nennender programmierbarer Taschenrechner (TI 59), der mir erlaubt, schnell die Daten verschiedener Varianten der Figur abzurufen.

Es ist (neben Größe- und Lageparametern) ein einziger Parameter, der die Form der Drachenstele bestimmt; eine Form, die zwischen den Extremen eines geraden Dreieckprismas und eines flachgedrückten, wendeltreppenartigen Gebildes schwanken kann.

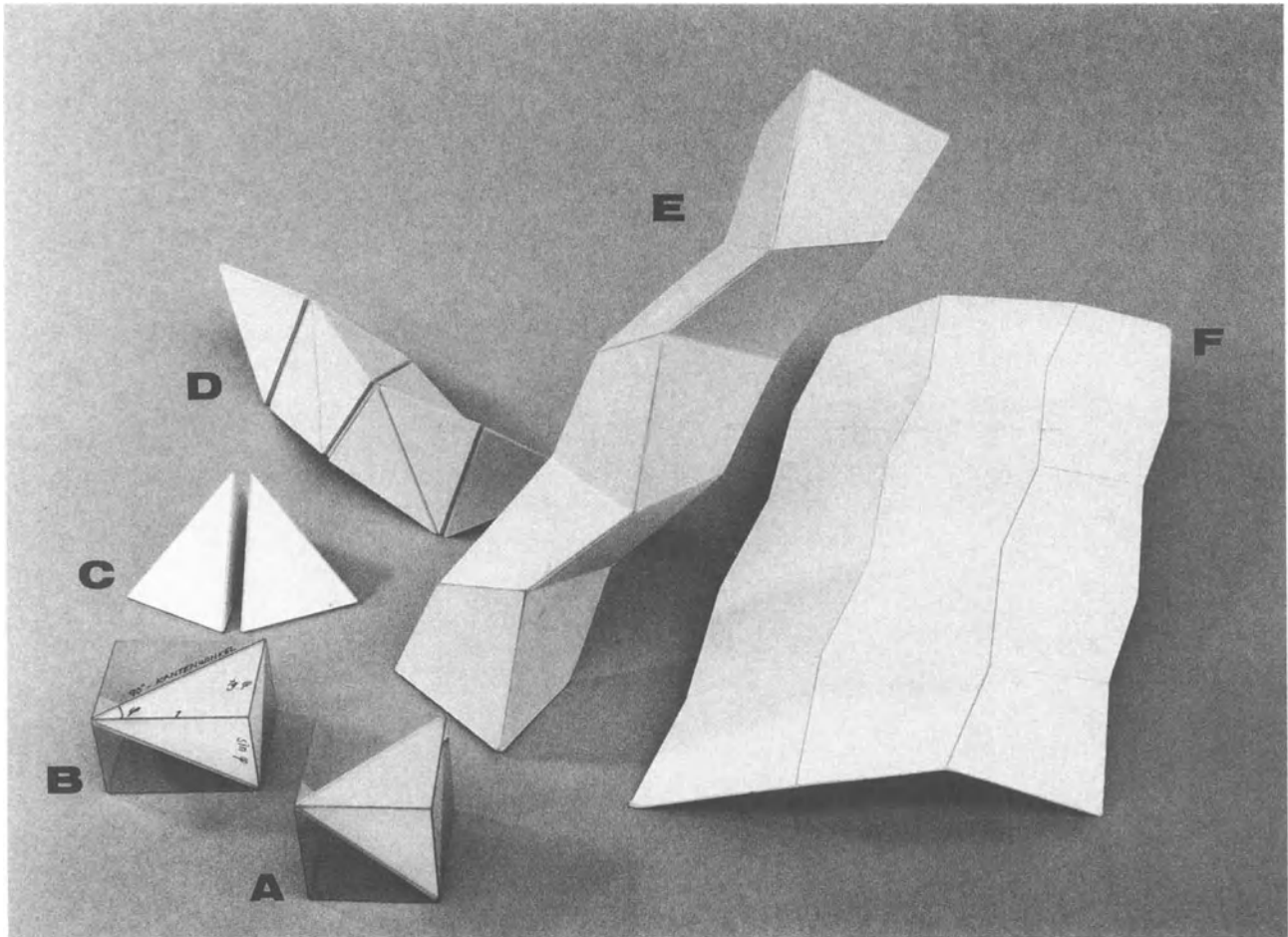
Aus den vielen Varianten, die ich als Kartonmodell vor mir habe, wähle ich jene aus, die mir am spannungsvollsten erscheint. Erst nachträglich erkenne ich, daß ich mich für eine hochspezielle Variante entschieden habe, der ein komplexes Proportionalsystem zugrunde liegt. Die Basis dieses Proportionalsystems ist die Zahl 1.553773974.

Zurück ins Kunsthaus: Aufnahme der genauen räumlichen Gegebenheiten; mein Traumbild in diesen Raum hineinprojizieren. Ich stelle mir vor: eine Figur, die von einem bestimmten Standort aus betrachtet dem Bild meiner regulären Drachenstele entspräche, in Wirklichkeit aber nur ein räumlich verzerrtes Abbild davon wäre. In Gedanken umschreite ich diese Figur, erkenne ihre perspektive Täuschung, male mir ihre aus jedem Blickwinkel sich verändernde Erscheinung aus.

Nun kommt CAD ins Spiel. Ich genieße freundlicherweise Gastrecht an der ETH Zürich (Architektur und CAAD bei Gerhard Schmitt). Am Computer modelliere ich die reguläre Drachenstele, skaliere sie, suche verschiedene Standorte, teste mögliche Blickpunkte. Wieder sind es rein intuitive Entscheide, die zu ersten Festlegung von Standort, Blickpunkt und Größe führen.

Damit ist erst das virtuelle Bild meiner Figur festgelegt, so wie es vom ersten Blickpunkt aus erscheinen soll. Die wirkliche Figur ist von diesem Blickpunkt aus deckungsgleich; jeder Punkt hat sich also auf dem Sehstrahl von mir weg oder auf mich zu bewegt. Die Formeln für diese Koordinatentransformation sind bald gefunden und erweisen sich als überraschen einfach.

$$V = \sqrt{\sqrt{2}+1} = 1.553773974$$



Entwicklung der regulären Drachenstele (Kartonmodelle).

A: Reguläres Tetraeder, symmetrisch dem Würfel einbeschrieben (Gleiche Kantenwinkel von $70^{\circ} 31' 44''$).

B: Analoge Anordnung eines isoedrischen Tetraeders im Quader mit dem Seitenverhältnis $1 : \text{tg } \zeta : \sin \zeta$ (ζ beliebig).

Dieses Tetraeder hat zwei gegenüberliegende rechtwinklige Kanten.

C: Zwei spiegelbildliche Exemplare dieses Tetraeders ergänzen sich zu einer Viereckpyramide. Ihre Grundfläche ist ein Drachenviereck.

D: Helixanordnung von alternierend links- und rechtshändigen Tetraedern.

E: Die je zwei in der gleichen Ebene liegenden Dreiecke der Figur D ergänzen sich zu einem Drachenviereck.

F: Die Abwicklung der Drachenstele ist identisch mit der Parkettierung der Ebene durch kongruente Vierecke. Durch gekoppelte

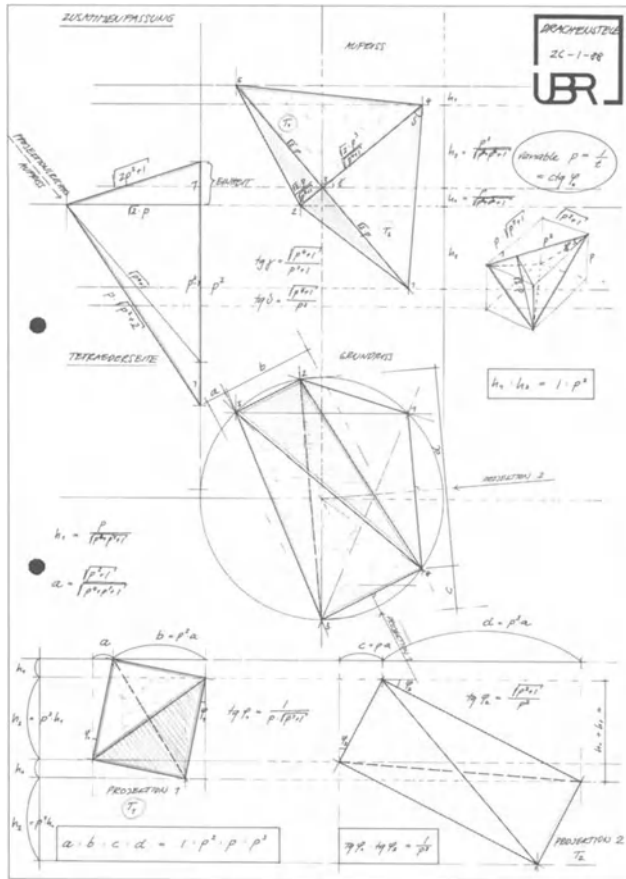
Diese Abbildung wird durch zwei Parameter erschöpfend beschrieben:

a = Abstand der Invarianzebene vom Augpunkt

b = Abstand der Perspektivebene (der nun endlichen Ebene aller Fluchtpunkte vom Augpunkt). Setzt man Augpunkt und Ursprung gleich, so wird jeder Vektor mit $r = b / (b - a + x)$ multipliziert. Ein kleines Lisp-Pro-

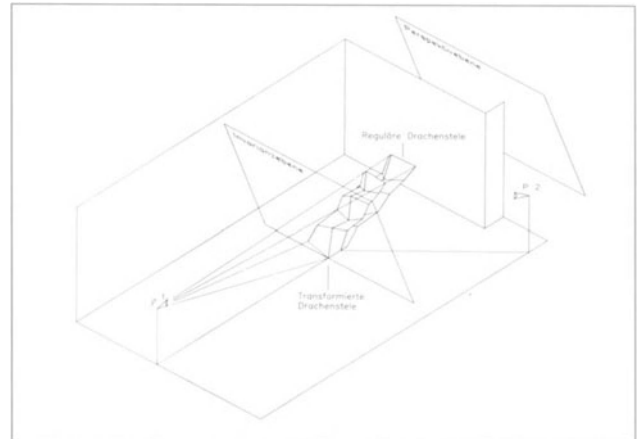
gramm erlaubt mir, diese räumlich perspektive Abbildung zu automatisieren.

Jetzt kann das Experimentieren beginnen. Ich teste die Wirkung dieser Abbildung zuerst an einfachen Körpern wie Würfeln und Quadern, um ein Gefühl für den Grad der erzeugten Verzerrung zu entwickeln, dann an der Drachenstele. Im Nu entstehen Varianten um Varian-



Parametrisierung der allgemeinen Drachensteile zur Auswertung mit dem TI-59. Die Form ist durch einen einzigen Parameter bestimmt ($p = \text{ctg } \zeta$).

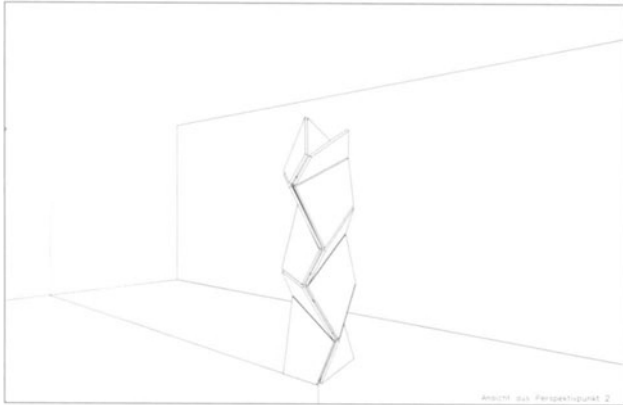
Die räumlich-perspektivische Abbildung der regulären Drachensteile führt zu einem neuen, nicht regulären Polyeder. Diese Zentralprojektion ist durch zwei Parameter definiert: Lage einer Invarianzebene und Lage der Fluchtebene.



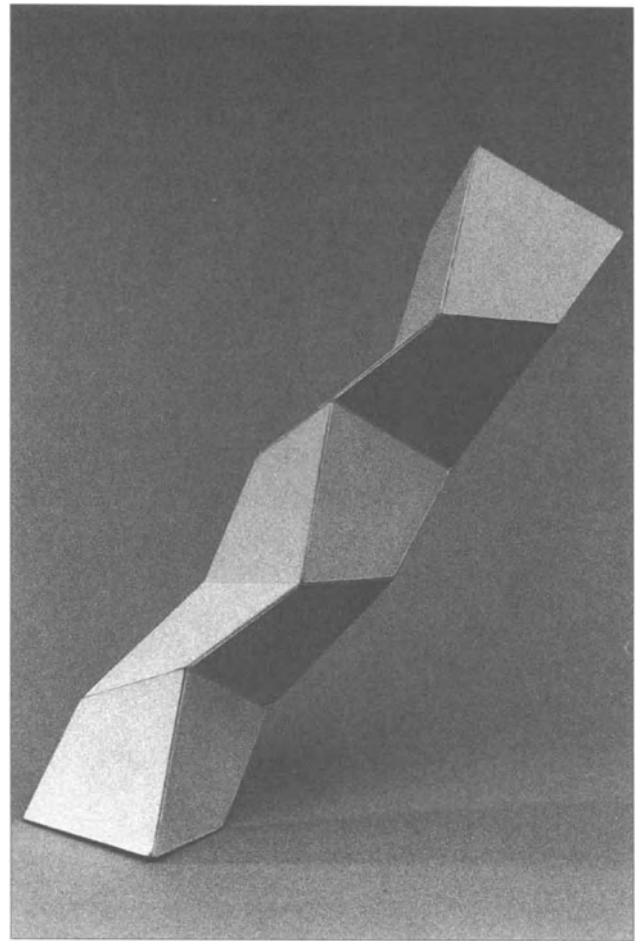
ten. Ich betrachte sie am Bildschirm simultan von vier verschiedenen Standorten aus, kann mir damit sehr bald eine Vorstellung ihrer räumlichen Wirkung erzeugen. Hier zeigt sich das Werkzeug Computer von seiner effizientesten Seite: undenkbar, diese Arbeit am Zeichenbrett mit den klassischen Mitteln der darstellenden Geometrie bewerkstelligen zu wollen. Eine einzige Variante, die ich am Computer in Sekundenschnelle generiere, würde in Handarbeit Wochen sinnloser Fleißleistung brauchen. Und was, wenn mich das Resultat nicht befriedigen würde?

Die Figur ist nun in ihren Umrissen geboren, aber noch nicht materialisiert. Stimmen Form und Proportion mit meinen ursprünglichen Vorstellungen überein? An der ETH stünden mir leistungsfähige Programme von Animation bis photorealistischen Darstellung zur Verfügung, um diese Frage zu klären. Ich benütze sie nicht. Als Plastiker mißtraue ich der süffigen Scheinwelt des animierten Bildes. Meinem Naturrell entsprechen viel eher: Ortswechsel, Rückzug in die stille Kammer, ein simples Kartonmodell bauen, lange schweigend davor sitzen, schauen. Ich brauche diese

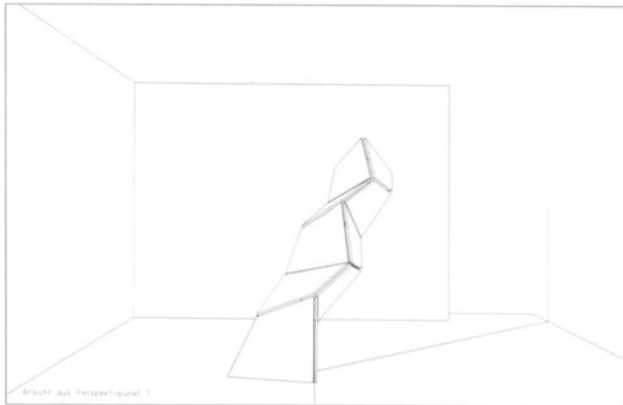
Ansicht aus Perspektivpunkt 2: Die perspektivische Verzerrung der Figur ist gegenkompensiert. Das Bild einer senkrechten Stele entsteht.



Reguläre Drachenstele mit $v = 1.553773974$ (Kartonmodell)



Ansicht aus Perspektivpunkt 1: Die Figur erscheint deckungsgleich mit der ursprünglichen regulären Drachenstele.



kontemplative Haltung, um meine Arbeit beurteilen zu können; bin froh, der nervösen Umgebung des Bildschirms entfliehen zu können.

Der nächste Arbeitsschritt führt mich zurück an den Computer: Die abstrakte Form in Werkpläne 1:1 umsetzen. Aus den 16 Polygonen werden Platten mit Materialdicke, eingeschnittenen Kanten, verbunden durch Stiftscharniere, die auf die Faltbarkeit der ursprünglichen Figur verweisen. Die gewählte Konstruktion erfordert ein Höchstmaß an Präzision, da die Scharniere nur ganz kleine Maßtoleranzen aufnehmen

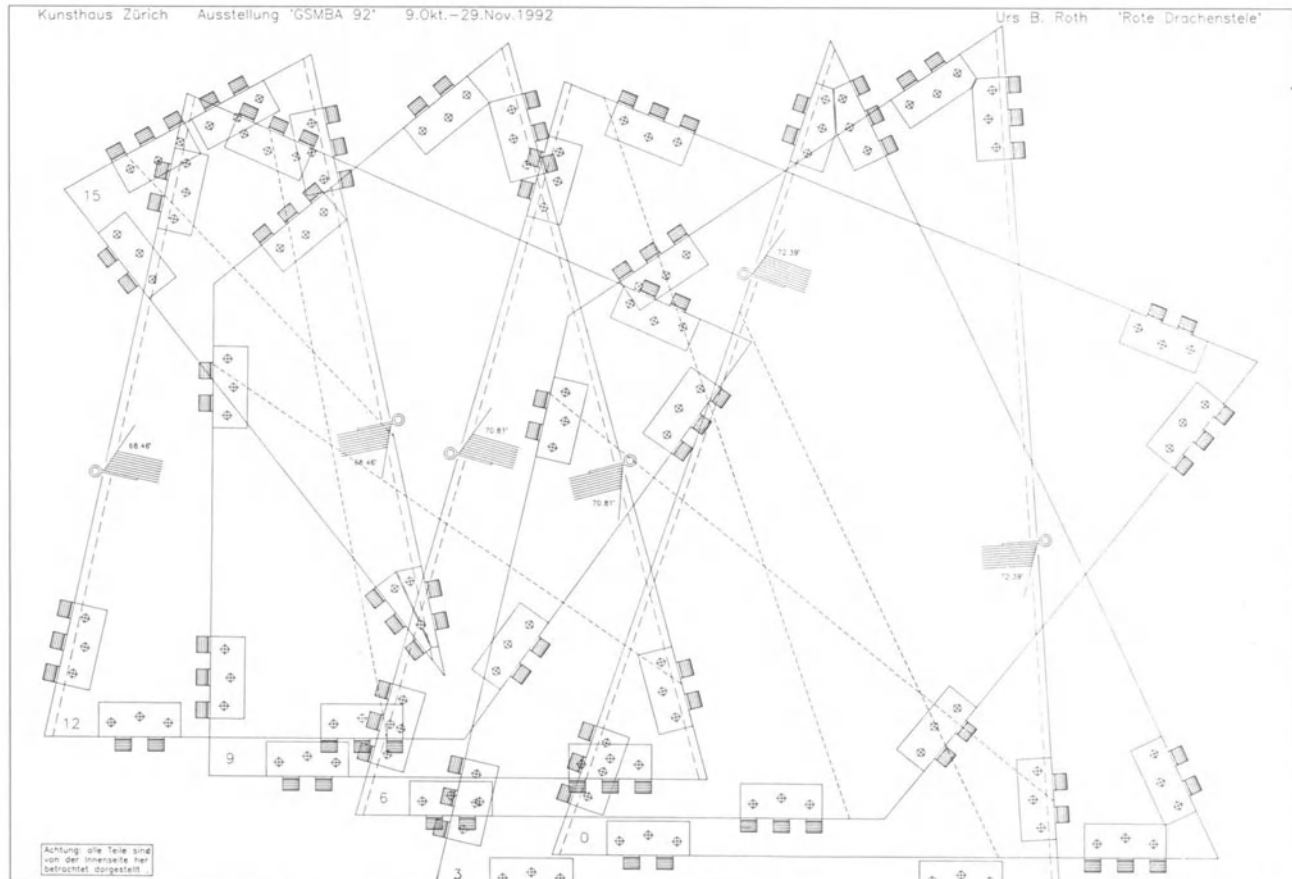
können. Nochmals: undenkbar, diese Bedingungen ohne CAD erfüllen zu können.

In der Werkstatt werden meine - notabene unvermaßten - Pläne 1:1 in Schichtholz umgesetzt. Und hier, am Rohbau der Figur, nehme ich die letzte Korrektur vor: Ein offenes Ende anstelle des ursprünglichen dreieckigen Abschlusses verdeutlicht den helixförmigen Aufbau der Figur, den man sich beliebig fortgesetzt denken könnte.

Betrachten wir das fertige Werk: Ein anscheinend ganz simples Ding. Ein gutes Dutzend, durch Scharniere



Konstruktionsdetail. Die Scharnierverbindungen verweisen auf die Faltbarkeit der ursprünglichen regulären Drachenstele.

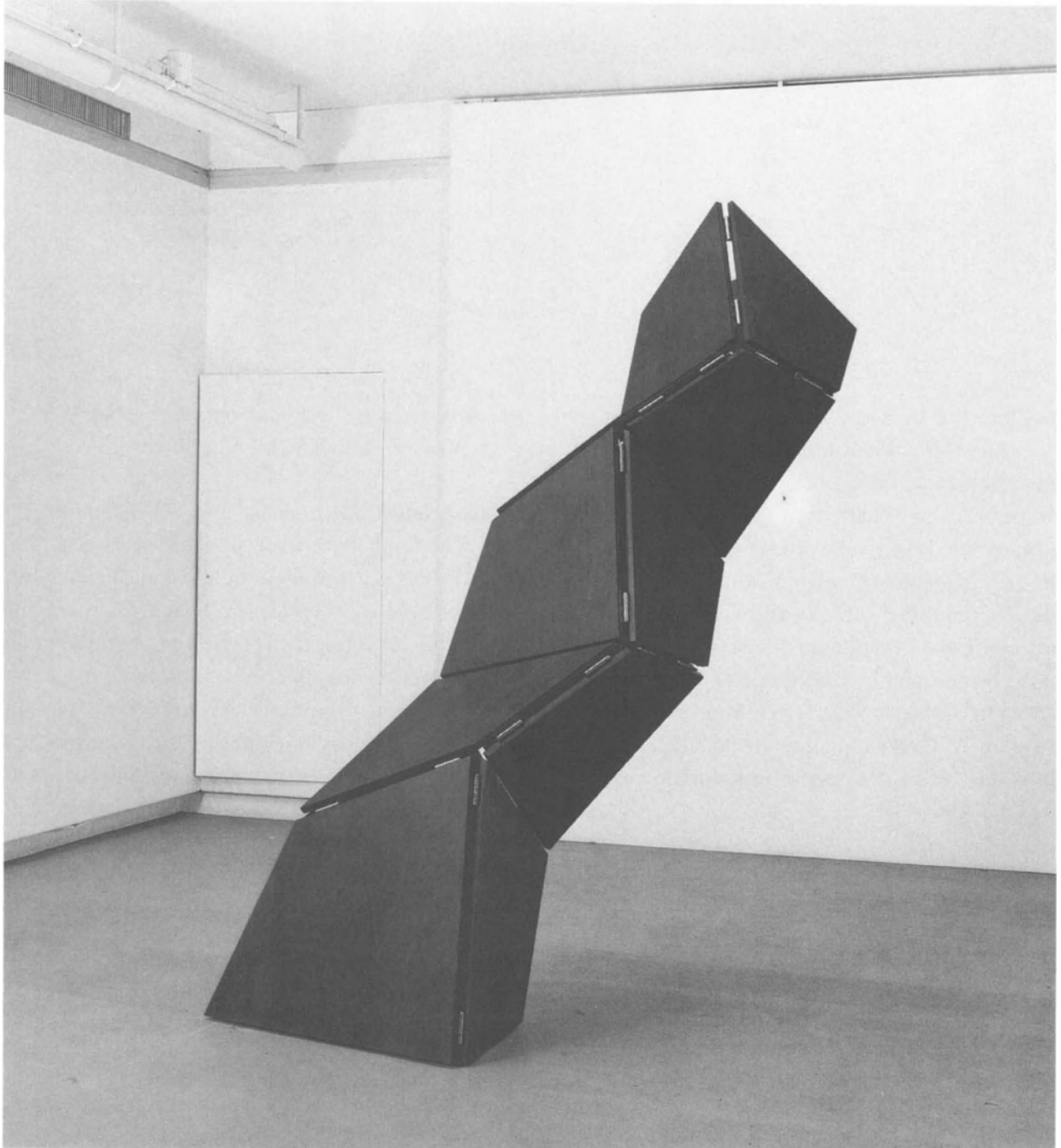


'Schnittmusterplan'. Um die notwendige Genauigkeit zu gewährleisten, wurden alle Teile im Maßstab 1 : 1 ausgeplottet und in der Werkstatt direkt in Schichtholz umgesetzt.

verbundene, rote Holzplatten, zu einem Turm aufgerichtet. Skulptur aus dem Computer? Nein, das wäre ein großes Mißverständnis. Sicher war in diesem Arbeitsprozeß der Computer ein unentbehrliches Konstruktionswerkzeug. Aber gerade an den entscheidenden Schritten war er nicht beteiligt: weder an der Konzeption, noch an der Selektion der Figur. Ich glaube, daß sich in diesen zwei Bereichen Dinge von unbe-

schreiblicher Komplexität in unseren Köpfen abspielen, Prozesse, die sich gegen eine formale Erfassung sperren. Als Vertreter der konkreten Kunst bin ich auf der Suche nach formal einfachen Objekten. Um so mehr freut mich das Wissen um die hintergründige Komplexität meiner Werke. Ich wünsche mir, der Betrachter spüre etwas von dieser geheimnisvollen, verborgenen Welt.

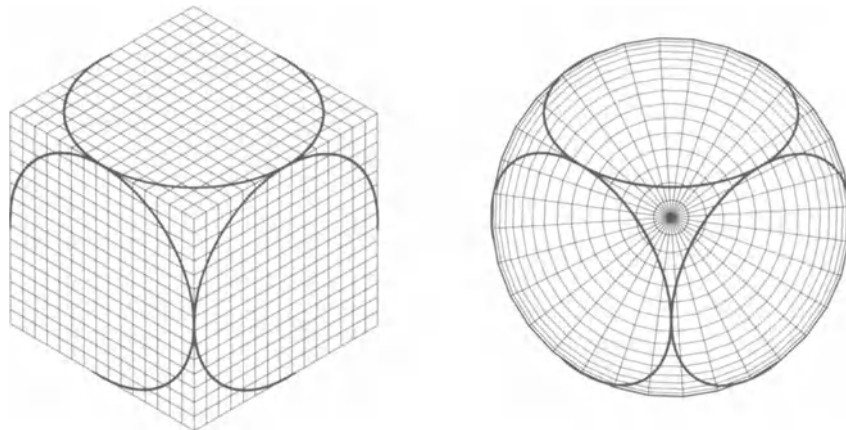
Rote Drachenstele. Skulptur aus 16 gelenkig
verbundenen Schichtholztafeln (69 x 179 x 230 cm).
Kunsthhaus Zürich 1992



Kraftstraße 35

Installation: Christoph Rütimann

Mitarbeit und Text: Florian Wenz



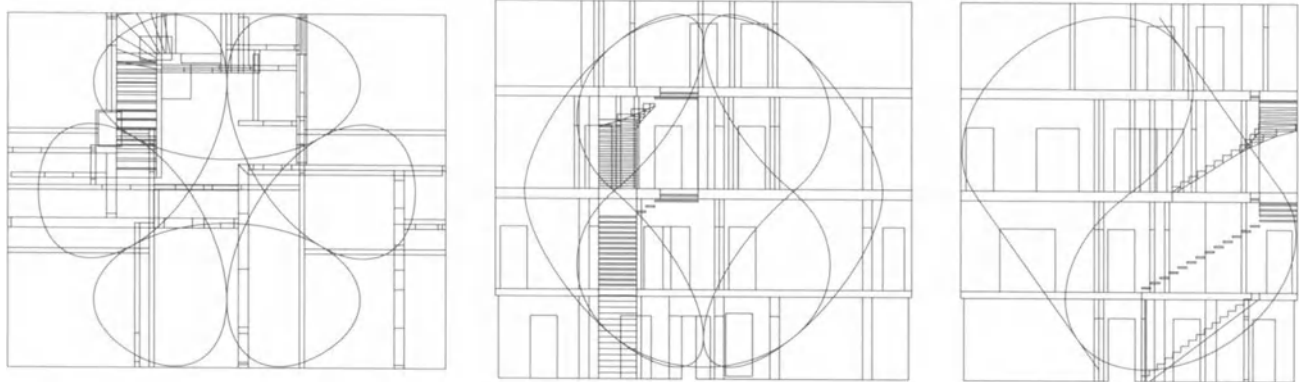
Die Entwicklung der Form aus der Verschneidungslinie von Kugel und Würfel und deren Platzierung im Haus Kraftstraße 35 in Grundriß und Ansichten

Diskurs: Die in den Computerwissenschaften entwickelten Methoden und Instrumente zeigen ihre Brauchbarkeit in der Anwendung auf den kreativen künstlerischen Schaffensprozeß. Wie alle anderen technischen Werkzeuge wird der Computer Teil unserer Kulturgeschichte, indem wir ihn in den kreativen Prozeß einbinden. Als verlustfreier präziser Mechanismus ist die Computermaschine neutral und führt ihre Operationen im unbegrenzten Datenraum aus. Als übergeordnetes Meta-System kann sie uns helfen, Strukturen zu erkennen und die materielle Welt mit Inhalten zu besetzen, die uns vorher verborgen waren.

Kooperation: Die Installation Kraftstraße 35 wurde von dem Luzerner Künstler Christoph Rütimann in Zusammenarbeit mit der Professur für Architektur und CAAD verwirklicht. Besonders in der Kooperation von Computer-Know-How und künstlerischem Vermögen können sich an der konzeptionellen Idee Ergebnisse auskristallisieren, die fachübergreifende Relevanz für beide Gebiete haben. In diesem Fall war die Konstellation besonders günstig, da auf der ande-

ren Seite ein Künstler stand, der mit eigenen Worten "die Wissenschaft als Steinbruch" benutzt.

Raumzeichnung: Zeichnen ist eines der Hauptanliegen von Christoph Rütimann, und das wichtigste Element seiner Zeichnung ist die Linie, die er auch mit anderen Medien, wie Waagen, Tönen, Photographie und in seinen Performances erforscht. Linie will hier verstanden sein als Spur einer Aktion, wie die Bewegung des Arms beim Zeichnen mit Tusche oder, in diesem Fall, der Verschneidung mehrerer geometrischer Elemente. Die Leseart von Linien als Abstraktionen ist abhängig vom Wahrnehmungsraster des Betrachters. Als Gerade oder Kurve vermittelt sie die Dynamik ihrer Entstehung, perspektivische Linien erzeugen Raum, Kreissegmente oder rechtwinklige Linien verweisen auf unendlich viele entsprechende Körper, als Projektion ist sie Schatten oder Abdruck eines Objekts. Schon die Linie selbst wird in ihrer Ausführung oder Materialtextur zum Ereignis. Die endlose Stahllinie ist Raum, Fläche, Linie und Material zugleich, sie beinhaltet die an ihrem Entstehungsprozeß beteiligten Körper und deren geometrische Wur-



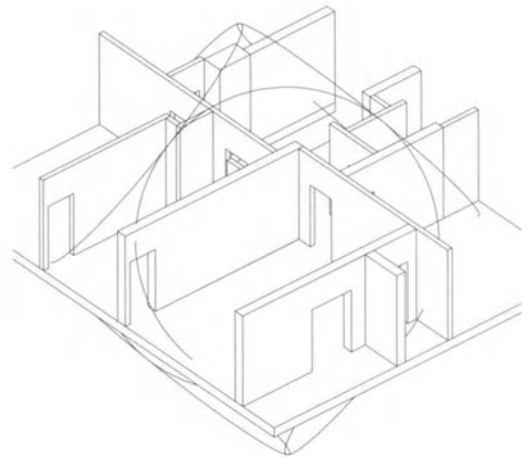
zeln. Auch Körper im abstrakten dreidimensionalen Datenraum des Computers vermitteln sich durch Linien aus farbigem Licht auf dem Monitor, allerdings ohne hier Spuren zu hinterlassen.

Objekt: Die endlose Linie im Raum entsteht aus der Verschneidungslinie von Würfel und Kugel. Sechs identische tropfenförmige dreiviertel-Kreis-Segmente sind in 90-Grad-Winkeln so zueinander angeordnet, daß sie die Form eines Würfels mit abgerundeten Ecken umschreiben. Diese Segmentflächen verhalten sich zueinander punktsymmetrisch, und alle Punkte auf der Linie sind gleich weit vom Mittelpunkt entfernt. Die geometrische Komplexität der Form findet Ausdruck in deren zweidimensionalen Projektionen und Schattenwürfen, aus denen sich kaum auf die dreidimensionale Form schließen läßt, und die aus bestimmten Projektionswinkeln überraschende Symmetrien aufweisen. Die bei der Installation verwendete Ausführung hat die Dimensionen 8m x 8m x 8m, und ist aus 12 identischen, Dreiechtel-Kreis-Segmenten zusammengesetzt, die aus Stahlrohr mit einem Durchmesser von sechs Zentimeter gebogen sind.

Arbeitsprozeß: Zunächst wurde das Innere des Hauses in der Kraftstrasse 35 mit einer Genauigkeit von etwa fünf Zentimetern in seinen relevanten Bauteilen vermessen. Auf Grundlage dieser Angaben wurden von Haus und Figur virtuelle dreidimensionale Modelle erstellt, die miteinander überlagert werden konnten. In Echtzeit-Visualisierung, Rendering und Projektion wurden unterschiedliche Rotationen und Positionen der Figur im Haus begutachtet, wobei auch die Lage der Rohre in den einzelnen Räumen und Konflikte in der Ausführung antizipiert werden konnten. In der endgültigen Konstellation wurde die Figur so mit dem Haus verschränkt, daß der immanente Würfel, in Grundriß und Schnitt diagonal gedreht, in der Raummitte des Hause zu liegen kam. Zur Ausführung wurden Werkpläne als orthogonale Aufsichten auf Wände und Decken im Maßstab 1:50 hergestellt, die die Schnittpunkte der Linie mit den Bauteilen zeigten, wobei auf eine Vermaßung wegen der Präzision der Ausdrucke verzichtet werden konnte. Diese Schnittpunkte wurden im Haus angezeichnet, die Durchbrüche angelegt und die Segmente der Form eingeschoben und miteinander verbunden.



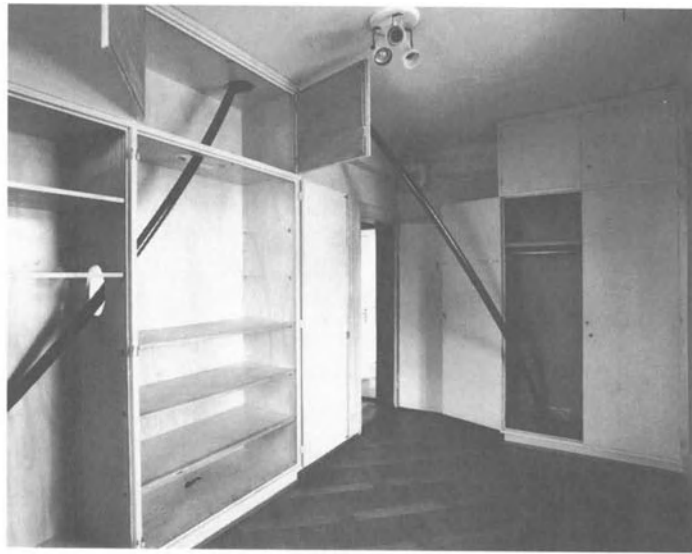
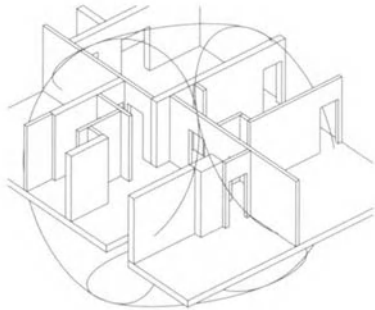
Durchdringung der Stahlrohrlinie mit dem Baukörper, axonometrische Ansicht des Obergeschoßes und ein Durchstoßpunkt mit freigelegten Materialschichtungen



Installation: Das Haus Kraftstrasse 35 stammt aus der Jahrhundertwende und liegt am Zürichberg in einem der repräsentativsten Wohngebiete der Stadt. Vor dem Abriss des Gebäudes und Umnutzung des Grundstücks gab der kunstverbundene Eigentümer dem Künstler die Gelegenheit, die Installation durchzuführen. Die dabei verwendete Stahlrohrlinie wurde vorher bereits zweimal bei Installationen in Perugia und Biel verwendet.

Bei Begehung der Installation fällt zunächst auf, daß die beiden, so unterschiedlichen und nicht für einander hergestellten Objekte fast wie selbstverständlich

zueinander stehen. Die Ausdehnung des Stahlobjekts im Raum paßt bis auf wenige Zentimeter exakt ins Innere des Hauses, in manchen Räumen werden Symmetrieachsen der vorhandenen Architektur aufgenommen. So wie die Figur aus der Verschneidung von Würfel, Kugel und Linie abgeleitet ist, verschneidet sich diese mit dem Inneren des Hauses. Die Kugel als Symbol der externen Welt hat hier ihr rekursives Gegenstück im Innenraum eines Wohnhauses als Welt der Bewohner. Da von keinem Standpunkt aus die geschlossene Linie ganz zu sehen ist, entsteht ihre Form aus der Interpretation der allgegenwärtigen Kreissegmente.



Verlauf der Durchdringungslinie über Keller-, Erd- und Obergeschoß, alle Photographien von Christoph Rütimann

Forschung und Machina

Ziele für die Forschung

Die Forschung nimmt im Computer Aided Architectural Design eine zentrale Stelle ein. Sie umfasst die Definition, das Testen und die prototypische Implementation zukünftiger Richtungen. Wie der Entwurf, so entwickelt sich auch die Forschung sowohl entlang vordefinierter Wege als auch in chaotischen Sprüngen. Beide Möglichkeiten garantieren erst das Entstehen wirklich neuartiger und nützlicher Programme.

Im folgenden beschreiben wir verschiedene Ansätze, die aus der Definition einer Forschungsrichtung mit dem Ziel funktionierender Prototypen entstanden. Zum Einsatz kommen dabei alle bisher bekannten Methoden und Instrumente. Im Sinne der Forschungsrichtung, welche die Unterstützung und Verbesserung des Entwurfsprozesses und seiner Ergebnisse zum Ziel hat, beschreiben wir Instrumente und konzeptionelle Instrumente. Für diese beiden Richtungen spielt die Art der Maschinen eine untergeordnete Rolle. Ansätze im Sinne der Forschungsrichtung, menschliches Verhalten zu imitieren und das menschliche Entwerfen auf einer Maschine zu re-kreieren, bezeichnen wir als kognitive Modelle. Die Art des Computers ist dabei von besonderer Bedeutung; man geht - Beispiel Neuronencomputer - bis zum Versuch des direkten Nachbaus der Struktur des menschlichen Gehirns.

Praktische Methoden und Instrumente

Darunter verstehen wir Methoden, die den Entwurfsprozeß, Teile davon oder sein Ergebnis verbessern oder erleichtern. Fortschritte in der Verfeinerung und Erweiterung bestehender Technologie sind besonders im Entwurf mit multimedialer Unterstützung zu erwarten, bei dem die Möglichkeiten der Verknüpfung von Graphik, Text, Bewegung und Ton ausgeschöpft werden. Für

Behörden und ausführende Betriebe werden ausgereifte Expertensysteme wichtig werden, die in genau abgegrenzten Wissensgebieten Unterstützung bieten. Sie sind auch für Entwerfende in der Form von Echtzeit-Entwurfskritik-Programmen interessant. Einige Beispiele sind:

Regelbasierte Systeme. Ein erster Schritt in diese Richtung ist ein regelbasiertes System, das vom Entwerfenden interaktiv und graphisch programmierbar ist [Flemming 1988a]. Dabei sind sowohl die Formen als auch die verknüpfenden Regeln frei definierbar - ein wesentlicher Fortschritt gegenüber bestehenden Systemen, bei denen Regeln und Formen jedesmal neu im Quellencode des Programms beschrieben werden müssen.

Parametrisierung. Im Maschinenbau sind seit einiger Zeit vollständig parametrisierte Programme in Anwendung. Die Parameter werden meist über Spreadsheets eingegeben, womit die Kombination eines traditionellen Mittels (wie der Zeichnung) mit einem neuen Mittel (wie dem Tabellenkalkulationsprogramm) entsteht. In der Architektur haben bereits verschiedene Programme Möglichkeiten dieser Art integriert: die skalierbaren Elementbibliotheken sowie die treppenbauenden Funktionen verschiedener CAD-Programme sind Beispiele.

Prototypen. Manipulierbare Prototypen sind eine Entwicklung, die von der sich durchsetzenden objektorientierten Programmierweise begünstigt wird. Denkbar und in einfacher Form bereits vorhanden sind zum Beispiel Fenster als Objekte, die sich ihrer Umgebung anpassen. Technische Einrichtungen, Mobiliar und andere Platzhalter fallen ebenfalls in diese Kategorie. Von dort ist der Weg nicht weit zu komplexeren Objekten, wozu vorfabrizierte Gebäude mit allen notwendigen Installationen gehören. Wir machten die Erfahrung, daß der Einsatz komplexer Prototypen auf Dauer nur dann Vorteile gegenüber parametrisierten Objekten

hat, wenn sich Zwischenergebnisse leicht visualisieren und manipulieren lassen.

Fallbasiertes Schließen. Dazu gehört eine Datenbank von komplett beschriebenen und indizierten Fallbeschreibungen, auf die während eines Entscheidungsprozesses zurückgegriffen werden kann. Fallbasiertes Schließen für die Detaillierung von Beispielen (siehe Methode: Fallbasiertes Schließen) ist realistischer vorstellbar als die Anwendung auf komplette Gebäude. Die Methode sei am Beispiel eines Dachdetails für ein leicht geneigtes Dach erklärt: In einer Fall-Datenbank befinden sich zwei Beispiele, die mit dem gegebenen Problem einiges gemeinsam haben. Das erste Beispiel stellt ein Flachdachdetail dar, das zweite ein steil geneigtes Dach. Die herkömmlichen Cut-and-Paste-Methoden versagen hier. Eine Fall-Datenbank jedoch, aus welcher der Computer mit Hilfe von analogen Folgerungen aus dem Flachdachdetail und aus dem Detail des geneigten Daches die geeigneten, also semantisch wichtigen Werte aussucht und zusammensetzt, kann zu einer Lösung führen. Besser noch, nachdem das neue Detail bestimmt und eingebaut ist, ergänzt es die Fall-Datenbank und verschnellert beim Auftauchen eines ähnlichen Problems die Suche. Wird jedoch ein völlig neues Detail entwickelt, das sich im Laufe der Zeit bewährt, so wird dies ebenfalls der Fall-Datenbank zugeordnet. Diese wächst mit der Zeit zu erheblichem Umfang und wird eine wichtige Entwurfshilfe darstellen.

Konzeptionelle Methoden

Damit bezeichnen wir Methoden, die, stark beeinflusst von den Naturwissenschaften, analoge Interpretationen und Erklärungen in der Architektur erlauben. Sie zielen daher nicht auf eine Verbesserung des bestehenden

Entwurfsprozesses ab, sondern nehmen durchaus das Risiko eines Paradigmenwechsels in Kauf. Hierunter fallen die Forschungsexperimente, die sich mit der fraktalen Natur der Architektur beschäftigen (siehe Methode: Fraktale), ebenso wie Experimente mit genetischer Software, die unter anderem Mutationen simuliert. Dabei handelt es sich um eine bisher noch nicht wohldefinierte Klasse von Programmen, die Wachstumsprozesse in der Natur zum Vorbild haben. Viele Anwendungsbeispiele sind denkbar, wie die Planung einer neuen Stadt in einem Gebiet mit hohem Bevölkerungswachstum. Hier könnte ein Ansatz nützlich sein, in dem jedes Gebäude, jede Wohnung, jede Produktionsstätte wie eine lebende, intelligente Zelle behandelt wird, die sich mit ihrer natürlichen Umgebung und ihren Nachbargebäuden aktiv auseinandersetzen muß. Jedes Gebäude oder jeder Komplex braucht Licht, Luft, Erschließung und Lärmschutz. Die rekursive Simulation akzeptiert naturgegebene lokale Begrenzungen und erwartet die Einpflanzung einer Gebäudezelle, die mit entsprechenden Wachstumsinformationen ausgestattet ist. Danach kann die Wachstumssimulation beginnen, und die Gebäudezellen breiten sich, bestimmt durch die jeweiligen mitgetragenen genetischen Informationen, in Wellen aus. Anders als in der herkömmlichen Planung füllen sie jedoch nicht lediglich das ihnen gegebene Gelände, sondern beginnen von Anfang an Interaktionen zwischen den Zellen: Viele sterben ab, da sie nicht genug Licht, Sonne, oder Ruhe haben, an anderen Stellen mit besonders günstigen Konditionen bilden sich Konzentrationen. Werden diese zu stark, so beginnen auch hier Zerfallerscheinungen, und nur in wenigen Fällen bilden sich nach vielen tausend Simulationszyklen stabile geometrische Konditionen heraus, die mit sich und der Umwelt in Einklang stehen - zumindest nach den Kriterien, die bei ihrem Wachstum

berücksichtigt werden konnten. Es ist undenkbar, dermaßen komplexe Simulationen sowie die aus ihnen resultierenden Informationen und Erkenntnisse mit herkömmlichen Mitteln zu gewinnen. Und es ist definitiv besser, diese Situationen zu simulieren, anstatt die Experimente gebauter Architektur mit ihren Bewohnerinnen und Bewohnern zu machen.

Kognitive Modelle

Darunter verstehen wir Ansätze, die auf Einsichten in die Arbeitsweise von Entwerfern und des menschlichen kognitiven Apparats basieren. Ömer Akin hat die Anwendung kognitiver Modelle auf den Entwurf, basierend auf Herbert Simons richtungsweisenden Arbeiten, ausführlich beschrieben [Akin 1986]. Sowohl die traditionellen als auch die KI-Methoden reichen nicht aus, ein komplexes Entwurfsproblem kompetent im umfassenden menschlichen Sinne zu lösen. Dazu muß auf eine Gesamtsicht, das sogenannte Mental Image, übergegangen werden, womit ein Entwurf ganzheitlich betrachtet und durchgeführt werden kann [Pu 1990]. Doch gibt es einige erfolgversprechende Teilgebiete.

Maschinenlernen (siehe Methode: Maschinenlernen) erlaubt Programmen, ihre eigene Leistung zu verbessern. Bisher in Anlehnung an menschliche Lernmethoden implementiert, kann dadurch Wissen gewonnen und permanent gespeichert werden. Aus dem besseren Verständnis des menschlichen Entwerfens leiten sich Entwicklungen ab, die Entwerfende über das Routine Design heraus unterstützen und innovatives sowie kreatives Design fördern könnten. Verstehende und lernfähige Programme sind zu entwickeln, die Architekten im Entwurf folgen und mit ihnen in eine partnerschaftliche Beziehung treten können. Maschinenlernen

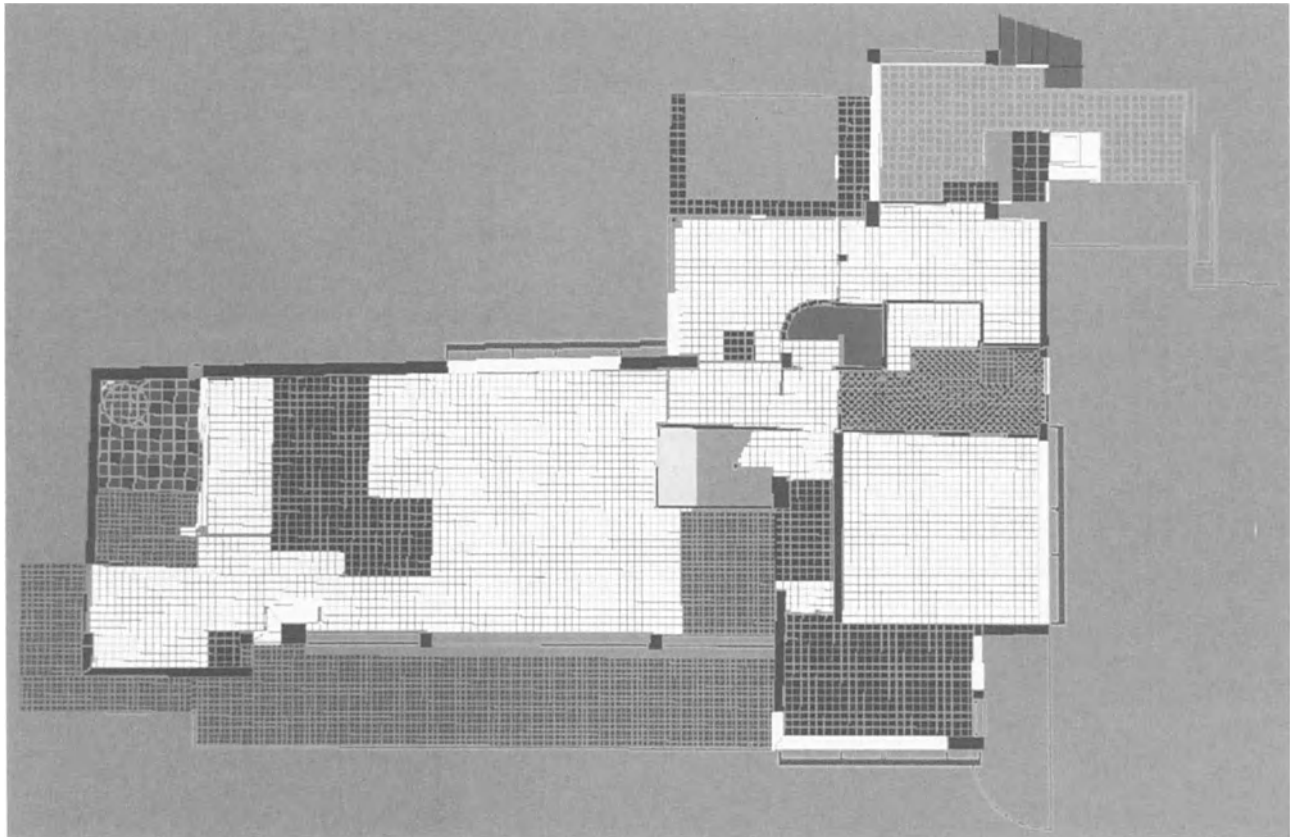
durch Deduktion, Induktion und Analogie sind bereits aktive Forschungsgebiete. Von Chen [Chen 1990] in jüngster Zeit teilweise implementiert, verspricht Maschinenlernen eine interessante Entwurfshilfe zu werden. Besonders im Bereich des graphischen Lernens ist viel Neuland zu erforschen. Hier hat sich bisher hauptsächlich die Methode des Lernens durch Vormachen (Learning by being shown) etabliert. Auf dem Gebiet des für die Architektur unentbehrlichen semantisch sinnvollen Maschinenlernens, gibt es noch keine erfolgreichen Projekte.

Instinktive Maschinen. Von menschlichen Fähigkeiten abgeleitet, entwickeln sich bereits Programme, die Instinkte und Reflexe simulieren. Sinn dieser Software ist die schnelle und instinktiv richtige Reaktion auf immer wiederkehrende und bekannte Situationen, zum Beispiel in Notfällen. Bestehende Applikationen in der Architektur sind nicht bekannt, doch denkbar. Daneben sind Computer mit Sensoren ausrüstbar, die menschliche Fähigkeiten ergänzen und in der Natur zu sehr guten Lösungen geführt haben, wie die Vermeidung schädlicher Einflüsse. Dazu gehören Ultraschall- und UV-Sensoren sowie das Erkennen gefährlicher Substanzen in der Luft.

Im folgenden werden Projekte in der Kategorie praktische Methoden und Instrumente vorgestellt. Case-Based Design beschreibt den gegenwärtigen Stand der Forschung auf dem Gebiet des fallbasierten Schließens in der Architektur sowie eine kleine Anwendung. Geodatenverarbeitung nimmt auf ein sehr konkretes Projekt zur Entwicklung neuer Entwurfs- und Analysehilfen Bezug. Das Projekt Eileen Gray beschreibt die anwendungsorientierte Forschung über die Rekonstruktion und Restauration der 'Maison en bord de mer' der Architektin Eileen Gray.

Wissenschaftliche Rekonstruktion: Eileen Grays Maison en bord de mer

Projekt: Stefan Hecker und Christian Müller
Leitung: Georg Mörsch und Gerhard Schmitt
Text: Stefan Hecker

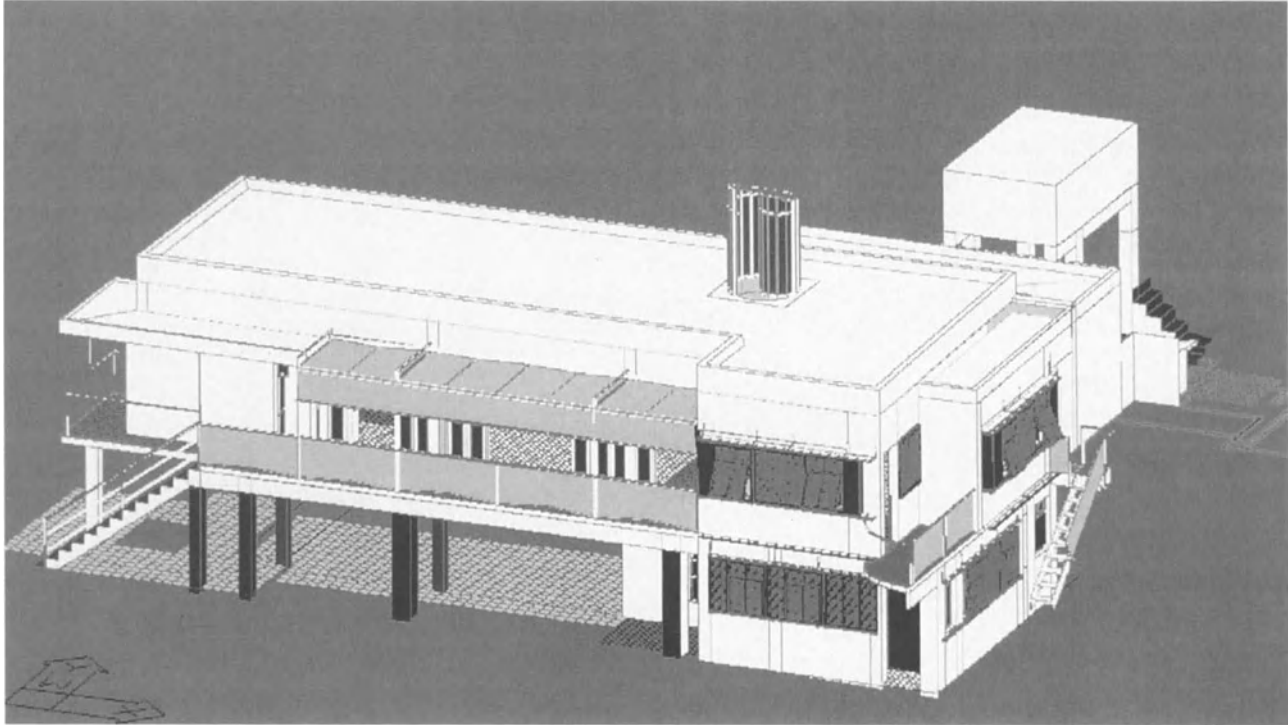


Hauptgeschoß. Räumliches Wechselspiel zwischen Begrenzungsmauern und unterschiedlichen Bodenbelägen. Hecker, 1993

Die 'Maison en bord de mer' ist ein wichtiges Baudenkmal der Moderne. Die Architektin und Designerin Eileen Gray erbaute es in den Jahren 1926 bis 1929 in Südfrankreich. Heute sind nur noch Teile von Gebäude und Einrichtung vorhanden, das Haus ist in einem schlechten baulichen Zustand und nicht zugänglich. Ebenso verhält es sich mit anderen kulturhistorisch wertvollen Bauten, die sich enormen Interessenskonflikten ausgesetzt sehen. Nicht nur kollidieren die Bedürfnisse der Allgemeinheit nach Erhaltung und Besichtigungsmöglichkeit mit den Intentionen der meist privaten Eigentümer, auch der

eigentliche denkmalpflegerische Umgang mit den Objekten ist oft von unterschiedlichen Auffassungen geprägt. Die Konzepte bewegen sich in einem Spannungsfeld, das von Konservierung über Restaurierung, Sanierung, Rekonstruktion bis hin zu einer Neuinterpretation reichen kann. Jeder Eingriff birgt die Gefahr, daß er aus einer anderen Sichtweise den Wert des Denkmals und dabei besonders seine geschichtliche Materie schmälert.

Das gemeinsame Forschungsprojekt des Instituts für Denkmalpflege und der Professur für Architektur und CAAD der ETH Zürich setzt sich zum Ziel, wissen-



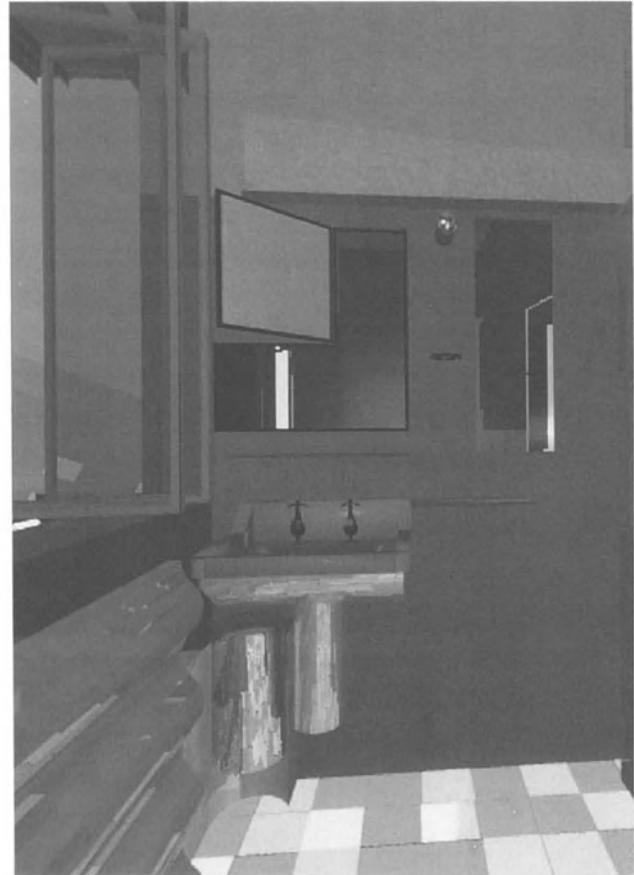
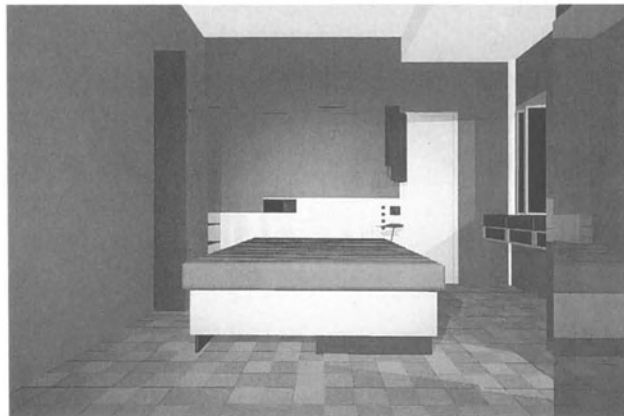
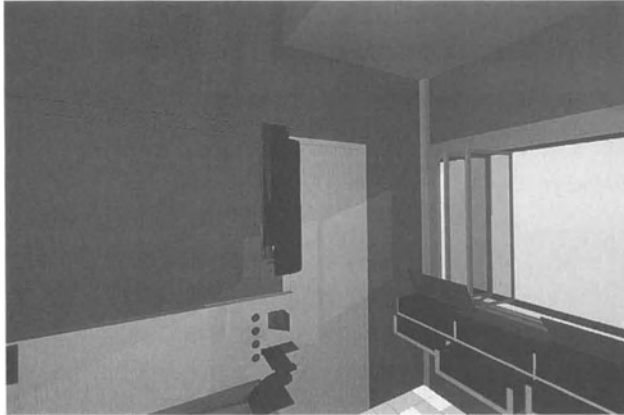
Axonometrie. Die verschiedenen Farben entsprechen unterschiedlichen Oberflächenattributen. Hecker, 1993

schaftliche Methoden für eine vom Artefakt losgelöste Auseinandersetzung mit dem Denkmal zu entwickeln. Es wird ein wirklichkeitsgetreues computerbasiertes Modell des Denkmals anhand der verbleibenden Spuren wissenschaftlich rekonstruiert und dessen Verwendbarkeit und Anwendungsregeln: (1) zur berührungsfreien, wissenschaftlichen Rekonstruktion der nicht mehr sichtbaren Teile, (2) zur detailgetreuen, realistischen Visualisierung des Originalzustandes, (3) zur didaktischen Vermittlung von Entwurfsgedanken, Entstehungsprozeß und baulichen Veränderungen, die im Laufe der Zeit erfolgten, sowie (4) zur Überprüfung der Auswirkungen von geplanten baulichen Eingriffen in ein Baudenkmal erforscht. Daß die 'Maison en bord de mer' für die Forschung ausgewählt wurde, hat, neben ihrem

historischen Wert, seinen Grund in der äußerst komplexen Raumstruktur. Erst die Überlagerung von Gebautem, Möblierung, Oberflächengestaltung und Farbgebung ergibt den Raumeindruck. Das Weglassen oder die Vereinfachung eines kleinen Details kann sich entscheidend auf den Raumeindruck auswirken. Das Objekt erfordert demnach eine äußerste Präzision bei der wissenschaftlichen Rekonstruktion und bietet die Gewähr, daß sich die gewonnen Erkenntnisse verallgemeinern und anwenden lassen.

Als Grundlage des Projekts dient eine umfangreiche Spurensicherung. Die noch vorhandenen Teile von Gebäude und Möblierung werden vermessen und photographiert, in Archiven wird nach Quellenmaterial gesucht, und Literaturrecherchen haben das Auffinden von reproduzierten Photos, Skizzen, Plänen

Photorealistische Darstellungen des Schlafraums. Hecker, 1993



und Beschreibungen zum Zweck. Die Spuren werden in einer Datenbank nach unterschiedlichsten Kriterien geordnet. Die Struktur der Datenbank erweist sich dabei für ein effektives Arbeiten als entscheidend. Dabei müssen der innere Aufbau des Objekts, die Art und Weise der angestrebten didaktischen Vermittlung und schließlich die verschiedenen Rahmenbedingungen der Informatik berücksichtigt werden.

Die eigentliche wissenschaftliche Rekonstruktion ist die Vereinigung aller Spuren im CAAD-Modell. Das bedeutet den präzisen Nachbau der Villa in Form eines Modells. Das detailgetreue Modell erlaubt den Vergleich des Quellenmaterials und somit den verlässlichen Nachvollzug der nicht mehr vorhandenen Elemente. Das CAAD-Modell muß aus einfachen Flächen aufgebaut werden. Nur so bietet es Gewähr

dafür, daß es für unterschiedliche Visualisierungsprogramme verwendet werden kann. Die Flächen müssen zudem nach Oberflächenbeschaffenheit und räumlicher Ausrichtung geordnet werden, um später Farben und Texturen präzise zuordnen zu können.

Die Visualisierung macht eine gezielte Auswahl von Teilen aus dem CAAD-Modell notwendig. Bestimmte Elemente des Modells müssen abstrahiert werden, um die Datenmenge in Grenzen zu halten. Farben und Texturen werden erzeugt und den Flächen zugeordnet. Die direkte visuelle Überprüfung erlaubt es nun, auch den nicht mehr sichtbaren Elementen, Oberflächenattribute zuzuweisen. Es werden möglichst realistische Renderings von geeigneten Ansichten erarbeitet und in einer weiteren Phase wird ein Video und eine interaktive Animation produziert.

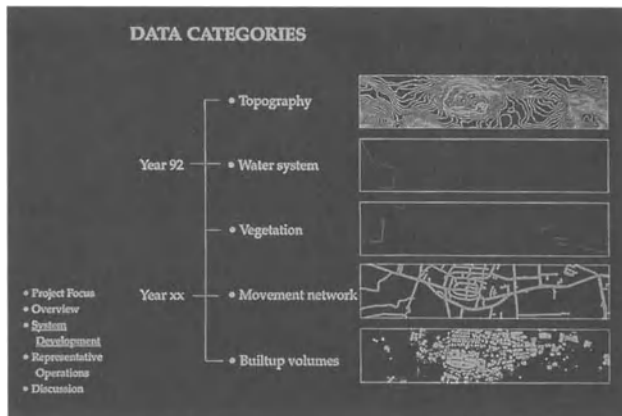
Geodatenverarbeitung

Projekt: Armin Grün, Franz Oswald und Gerhard Schmitt

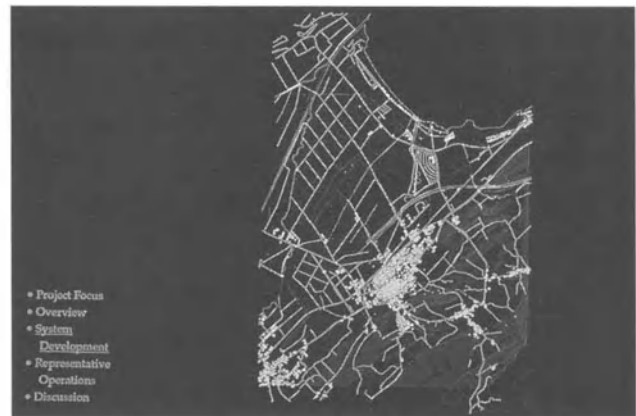
Mitarbeiter: Martina Meister, Frank Neumann

Programm: Bharat Dave

Datenkategorien, abgeleitet vom Themenraster der Gruppe Architektur. Dave, 1992



Zusammenstellung und Überlagerung aller Datenkategorien in einer einzigen Darstellung. Dave, 1992



Seit jeher faszinieren die Erforschung der Vergangenheit und die Planung der Zukunft. 1990 setzten es sich drei Professuren an der ETH Zürich zum Ziel, ein neues Planungs- und Entwurfs-Instrumentarium zu entwickeln: die systematische Erkundung einer Hard- und Software-Umgebung, in der Modelle urbaner Entwicklung repräsentiert, manipuliert und entworfen werden können. Das Schwergewicht liegt dabei auf der Identifizierung und der Darstellung von Informationen, die in Städtebau und Architektur für Analyse und Entwurfszwecke nützlich sind. Als Gegenstand eines ersten prototypischen Systems diente die Stadt Aventicum. Beschreibungen und Dokumente über die Stadt, Pläne in verschiedenen Maßstäben und aus verschiedenen Zeiten werden in einer Computer-Datenbank zusammengefaßt. Durch die Verwendung der Datenbank sind Entwurfs- und Wachstumsaspekte interaktiv erfahrbar.

Die drei Forschungsgruppen, die sich mit der Entwicklung neuer Methoden und Instrumente für Architektur und Planung befassen, sind Photogrammetrie, Entwurf und CAAD. Das Entwurfsteam erar-

beitet ein Themenraster zur Erfassung der topographischen Merkmale des Gebietes. Mit ihm werden Analysen erstellt. Die Aufgabe der Gruppe Photogrammetrie ist die Beschaffung der Geodaten und deren Strukturierung in einer Form, welche die Übernahme durch das CAAD-Team zur Weiterverarbeitung erlaubt. Das CAAD-Team agiert als Vermittler zwischen Entwurf und Photogrammetrie. Es übernimmt von der Photogrammetrie geometrische Rohdaten und entwickelt Konzepte und Programme, diese Daten für in Planung und Entwurf sinnvolle Informationen umzusetzen.

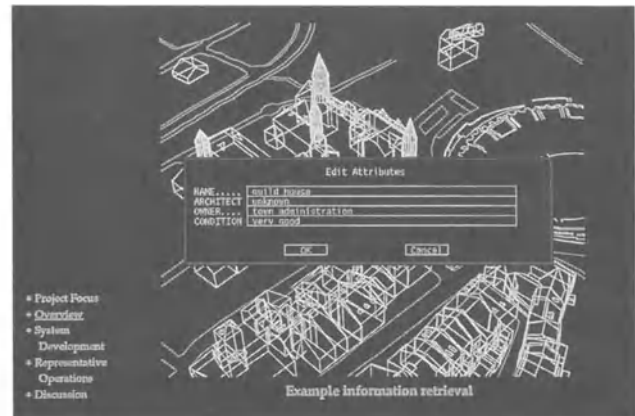
Für das CAAD-Team steht das Ziel im Vordergrund, eine benutzerfreundliche Analyse-, Planungs- und Entwurfsumgebung zu schaffen, bei der die wichtigsten Aktivitäten in einem einzigen Programm ablaufen können. Die Schritte dazu sind:

(1) Übernahme zwei- und dreidimensionaler Datensätze von der Photogrammetriegruppe. Dies stellte sich als schwieriger denn erwartet heraus, da die Daten in einer Form anfielen, die eine Umstrukturierung notwendig machte.

Erste Eintragungen in die Datenbank über dem Plan von Avenches. Programm: Dave, 1992



Abfrage der Datenbank aus dem dreidimensionalen Modell. Dave, 1992



(2) Verwandlung der Daten durch sinnvolle Gliederung und Hinzufügen von Attributen in Information. Dazu war die Entwicklung einer geeigneten Datenrepräsentation und von Datenbankstrukturen notwendig. Für das ausgewählte CAD-System AutoCAD wurde Datenbankfunktionalität implementiert. Das bedeutet, daß Anwendern innerhalb des CAD-Pakets SQL-ähnliche Abfragefunktionen zur Verfügung stehen. Diese Abfragefunktionen können graphisch und interaktiv benutzt werden.

(3) Herstellung notwendiger Entwurfs-, Datenbank- und Visualisierungs-Funktionalität. Dazu mußte das kommerzielle CAD-Programm mit CAAD-Kapazität ausgestattet werden. Dies bedeutet das Programmieren architekturenspezifischer Funktionen.

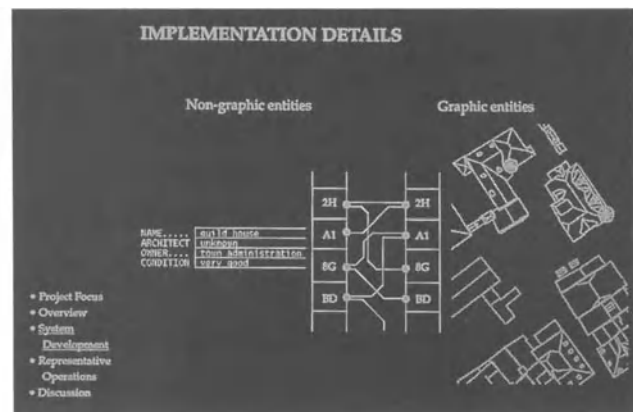
Diese Schritte sollen den Kern des Entwerfens und Planens unterstützen, ohne von den Anwenderinnen und Anwendern tiefgreifende Kenntnisse in Datenbanken oder Geo-Informationssystemen (GIS) zu erfordern. So wird ein bisher nicht vorhandenes Instrument geschaffen, das erlaubt, in Planung und Entwurf qualitative und quantitative Verknüpfungen

herzustellen. Das Projekt sieht die Integration von GIS-Funktionalität und Entwurfsinstrument vor. Hier besteht ein Konflikt: Einerseits ist die Funktionalität der meisten GIS-Systeme überwältigend, während zugleich wichtige dreidimensionale Entwurfswerkzeuge fehlen; andererseits bieten CAD-Programme gute Modellierfähigkeit, während ihnen GIS-Funktionalität fehlt. Zudem verlangt das Erlernen verschiedener GIS- und CAD-Systeme zuviel Zeit, und bei der Anschaffung verschiedener Systeme fallen die Software- und Wartungskosten schwer ins Gewicht. Zunächst versuchten wir, eine kommerzielle Software zu finden, die der Lösung dieser Aufgaben möglichst nahekommt. Dazu wurden kommerzielle GIS sowie neue CAD-Programme untersucht. Das Resultat dieser Evaluationen war, daß es zwar Programme mit vielen der geforderten Eigenschaften gibt, aber daß keines der untersuchten Programme die gesamte Funktionalität aufweist und sich außerdem in etwa 12 Stunden Vorlesungen und Übungen so vermitteln läßt, daß damit praktische Entwurfsarbeit zu leisten ist. Die Software-Evaluation wird jedoch in regelmäßigen

Nebeneinander von Vektordaten und Rasterdaten.
Beide Repräsentationen sind zu Entwurfszwecken kombinierbar



Verbindung graphischer und nicht-graphischer Objekte.
Graphische Darstellung der Attribute. Dave, 1992



Abständen im Verlauf des Projekts erneut durchgeführt. Seit Beginn stand auch zur Diskussion, ein kommerzielles relationales oder objektorientiertes Datenbanksystem mit dem CAD-System zur Verwaltung der großen anfallenden Datenmengen zu koppeln. Der Grund für die jetzige Lösung, essentielle Datenbankfunktionalität im CAD-System zu schaffen, liegt in der Absicht, oft benötigte Funktionen möglichst elegant in eine Entwurfsumgebung zu integrieren sowie die Zahl kommerzieller Programme für das Projekt so gering wie möglich zu halten. Die bisher erreichte Struktur erlaubt die spätere Einbindung kommerzieller Datenbanken.

Von den vielen Möglichkeiten, welche die Forschungsergebnisse des Projekts bieten, sind für die Studierenden besonders folgende Aspekte von täglichem Nutzen:

(1) Digitales Geländemodell. Es ergibt sich aus den Daten der Photogrammetriegruppe und kann durch

Triangulierung oder Überziehen mit einem Netz von Quadraten zur einfachen Darstellung von Landschaft verwendet werden.

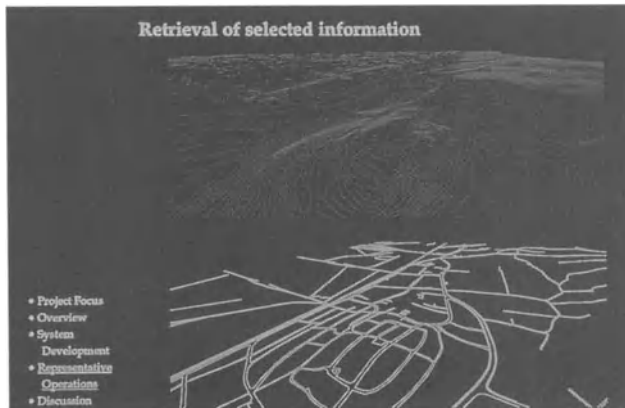
(2) Themenraster. Die wichtigsten Aspekte des Planungsgebietes finden sich hier. Beispiele sind das Straßennetz, die Parzellierung, Gebäudetypen, Vegetationstypen und Wasserläufe. Durch die große Informationstiefe des digitalen Modells sind alle zum Themenraster gehörende Informationen stets verfügbar.

(3) Datenbankabfragen. Damit lassen sich wichtige Zusammenhänge zwischen graphischer und nicht-graphischer Information erkunden.

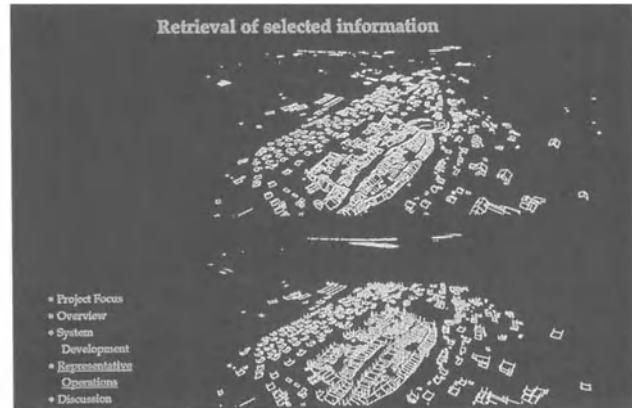
(4) Dreidimensionale Entwurfswerkzeuge. Sie stehen als Basisfunktionen im CAD-Programm zur Verfügung und wurden von uns um viele Zusatzfunktionen erweitert.

(5) Real-Time-Visualisierung. Dazu steht das an der University of Toronto entwickelte Polytrim-Programm zur Verfügung, das bei der Herstellung einfa-

Perspektivische Darstellung der Höhenlinien und des Straßennetzes. Daten Meister, 1992. Programm Dave, 1992



Herausheben ausgewählter Informationen. Daten Meister, 1992. Modell Dave, 1992



cher Videos große Hilfe bietet [Danahy 1992] und sich im Unterricht bewährt hat.

(6) Erzeugung von multimedialen Präsentationen. Sobald mehrdimensionale Modelle vorhanden sind, steht dem Entwurf multimedialer Präsentationen nichts mehr im Wege. So lassen sich zum Beispiel im Programm Showcase in kurzer Zeit effiziente Präsentationen zusammenstellen, die Text, statische Bilder, Animationen, Real-Time-Bewegungen und Ton einschließen.

Auch für die Praxis ist das Projekt von Interesse. Im Schweizer Kanton Zürich ist die Entwicklung eines geographischen Informationssystems geplant, in anderen Kantonen sind ähnliche Bestrebungen im Gang [Küng 1993]. Für Bauherren, Planer und Architekten ist es wichtig, Informationen aus GIS-Systemen so schnell und selektiv wie möglich auf ihre Maschinen zu übertragen. Da die wenigsten Architekturbüros heute oder in naher Zukunft mit GIS-

Systemen ausgerüstet sind, müssen Schnittstellen zwischen allgemein zugänglichen Datenbanken, digitalen Landschaftsmodellen und anderen Inhalten von GIS-Systemen entwickelt werden. Absolventinnen und Absolventen mit solcher Ausbildung werden in der Lage sein, Erfahrungen schnell in die Praxis umzusetzen. Daraus wird sich mittelfristig ein neues Arbeitsverhältnis zwischen Verwaltung, Planungs- und Architekturbüros entwickeln. In der Architekturpraxis besteht die Notwendigkeit eines integrierten Systems für Analyse und Entwurf ebenfalls. Allerdings sind hier die Anforderungen an Datensicherheit und Verlässlichkeit der Information weitaus höher. Die Anschließung einer kommerziellen Datenbank wird daher notwendig sein. Für ein schnelles konzeptionelles Entwurfsinstrument ist das vorgeschlagene System auch in der Praxis geeignet, da die zusätzlichen Programmierungen auf verschiedenen Hardware-Plattformen funktionieren [Dave 1992a]. Die



Visualisierung verschiedener Datenkategorien von Avenches und Umgebung. Dave, 1992

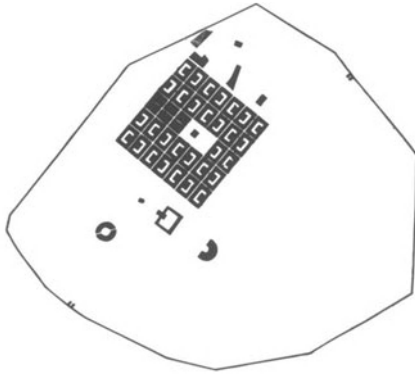


Visualisierung der mittelalterlichen Altstadt von Avenches als Drahtmodell. Dave, 1992

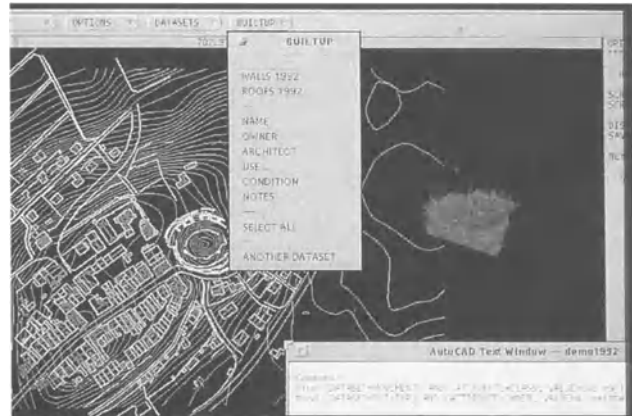
kommerzielle Weiterentwicklung der CAD-Software kommt dem Anliegen dieses Projekts sehr entgegen. So erlaubt das CAD-Programm, Flächen im Raum mit Bitmap-Objekten zu hinterlegen und darauf unsere Funktionen einzusetzen. Auch die Einbindung der CAD-Daten in kommerzielle Datenbanken wie Oracle hat sich wesentlich vereinfacht. Falls notwendig, lassen sich damit große Datenmengen verwalten, die nicht direkt während des Entwurfs zur Verfügung stehen müssen. Es ist absehbar, daß sich durch die Kombination neuer Funktionen in der kommerziellen Software und aus unseren eigenen Programmierun-

gen ein robustes und für die Praxis interessantes Paket entwickeln läßt. Die Verbindung eines CAD-Programms mit Datenbank- und entwurfsunterstützenden Funktionen hat gegenüber den bestehenden Insellösungen für die Einzelgebiete große Vorteile. Aus der Integration von Geo-, Architektur- und CAAD-Daten kann auch eine bessere Integration der Beteiligten am Planungs- und Entwurfsprozess hervorgehen. Mit dem so entstehenden Instrument kann die Entwicklung eines Gebiets von den ersten bekannten Anfängen bis in die Gegenwart verfolgt werden. Projektionen in die Zukunft werden möglich.

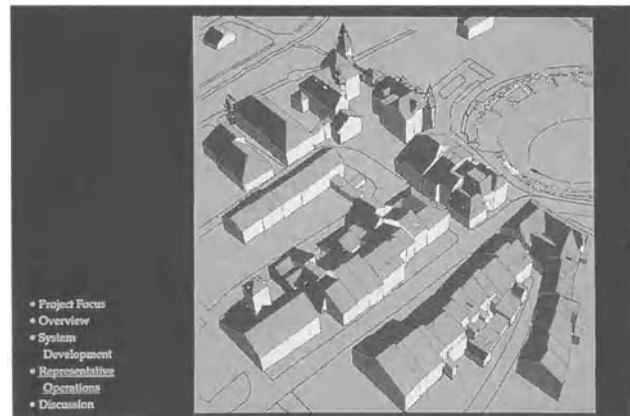
Aventicum in römischer Zeit mit rechteckigem Stadtgrundriß, Tempel, Theater und Stadtmauer



Blick auf ein Teilmodell des mittelalterlichen Avenches. Daten Meister, Modell Dave, 1992.



Avenches im Mittelalter. Die neue Stadt entwickelt sich um das ehemalige Amphitheater. Dave, 1992



Nachbearbeitung des räumlichen Drahtmodells mit einem Paint-Programm. Dave, 1992

Perspektivische Darstellung des dreidimensionalen Straßennetzes. Avenches im Vordergrund links, das Ufer des Murtensees oben links. Daten Meister, Visualisierung Dave, 1993



Fallbasiertes Entwerfen

Projekt: Boi Faltings, Ian Smith und Gerhard Schmitt

Mitarbeiter: Shen-Guan Shih, Kefeng Hua und Simon Bailey

Die Repräsentation und Aneignung von architektonischem Wissen sowie dessen Formalisierung für die Verwendung in Computerprogrammen ist noch immer ein schwieriges Forschungsgebiet, ohne dessen Lösung keine intelligenten CAAD-Systeme möglich sind. Traditionellerweise sind Produktionsregeln ein weitverbreitetes Paradigma, Wissen für Computeroperationen zu repräsentieren. Die Regeln und Aktionen können in sich einen hohen Komplexitätsgrad erreichen und durch Kontrollmechanismen, wie Vorwärtsdeduktion oder Rückwärtsdeduktion, miteinander verknüpft sein (siehe Instrument: Programmieren II - Wissensbasierte Programme). Für Anwendungen in Gebieten wie Architektur und Design ist es jedoch praktisch unmöglich, ein kohärentes Set solcher Wenn-Dann-Regeln zu formulieren, das den großen Reichtum von architektonischem Wissen adäquat umfaßt. Noch ist nicht bekannt, welche Art von Wissen notwendig ist, um einen kompletten Entwurf mit Regeln zu erzeugen. Außerdem gibt es starke Hinweise aus der Psychologie, daß beim Entwerfen nicht lediglich Regeln zum Einsatz kommen, sondern erfahrene Entwerferinnen und Entwerfer vermehrt auf zuvor gelöste, ähnliche Probleme zurückgreifen. Diese Beobachtung hat zum Paradigma des fallbasierten Schließens (Case-Based Reasoning) geführt, das einige Probleme der regelbasierten Systeme eliminiert (siehe Methode: Fallbasiertes Schließen).

Wir wenden fallbasiertes Schließen auf ausgewählte Entwurfsprobleme an, da diese Methode teilweise in der Architektur entstanden ist. Das Arbeiten mit Fällen ist für die Architekturforschung besonders in der Form des fallbasierten Entwerfens (Case-Based Design) interessant, das mit dem fallbasierten Schließen verwandt ist. Fallbasiertes Entwerfen geht davon aus, daß bei der Verwendung des Computers im Entwurf das

vollständige Neu-Erzeugen von Lösungen sehr unwirtschaftlich ist. Vielmehr wird versucht, neue Lösungen durch Adaptierung bestehender Objekte guter Qualität zu erreichen.

Als erstes Beispiel haben wir eine Wohnanlage von Anton Schweighofer in Wien und das Felder-Haus von Campi-Pessina gewählt. Diese Gebäude gliedern wir in eine große Zahl sogenannter Basis-Parameter. Räumliche Beziehungen und funktionelle Verknüpfungen sind in einer Graphen-Repräsentation festgehalten, die nach Eingabe des Gebäudes in eine spezielle AutoCAD-Shell automatisch generiert wird. Ändern sich nun die äußeren Bedingungen für das Gebäude, so findet eine Fall-Anpassung (Case Adaptation) statt. Dies bedeutet, daß das Programm zunächst nach Konflikten eines bestehenden Falles (Case) mit der neuen Situation sucht, diese Konflikte in Parameter umwandelt, danach die große Zahl der Parameter durch Dimensionalitäts-Reduktion (Dimensionality Reduction) auf wenige zurückführt und mit diesen den ursprünglichen Fall adaptiv verändert [Hua 1992]. Bei näherem Hinsehen laufen in den einzelnen Prozessen folgende Schritte ab:

(1) Evaluation des bestehenden Falles in der bestehenden und in der neuen Umgebung, um Dimensions- und topologische Konflikte zu finden. Dimensions-Konflikte lassen sich durch Dehnen oder Stauchen beheben, was aber in vielen Fällen nicht möglich ist, ohne wichtige Proportionen oder räumlich-funktionale Zusammenhänge (die Topologie des Falles) zu verletzen.

(2) Existieren Dimensions-Konflikte, so findet das Programm die entsprechenden Parameter und definiert ein Set von Restriktionen oder Constraints. Es addiert diese Constraints zu den bereits mit dem Fall gespeicherten Constraints. Dabei werden normative Con-

straints (formbezogen) und funktionale Constraints (funktionsbezogen) getrennt behandelt.

(3) Falls topologische Konflikte existieren, ruft das Programm Transformationsregeln auf, welche die topologische Abstraktion des Gebäudes, einen Graph, so anpassen, daß die ursprünglichen Funktionen beibehalten werden. Nach den topologischen Anpassungen baut das Programm die Dimensions-Constraints, die bei der Manipulation des Graphen gelockert werden mußten, wieder auf.

(4) Unter Benutzung derjenigen Constraints, die in der neuen Umgebung nicht erfüllt werden, sowie von Constraints in der lokalen Abstraktion des Entwurfs, definiert das Programm alle Parameter in diesem lokalen Constraint-Netzwerk mit Anpassungs-Parametern durch einen prozeßunabhängigen Dimensionalitäts-Reduktionsprozeß.

(5) Das Programm variiert danach die Anpassungs-Parameter, um sicherzustellen, daß keine Constraints verletzt sind. Es leitet die Änderungen an die entsprechenden Dimensions-Variablen in der Gebäudebeschreibung weiter.

(6) Das Programm überprüft den Erfolg der Anpassung. Falls keine Constraints verletzt sind, ist der Prozeß erfolgreich abgeschlossen.

(7) Falls Constraints verletzt werden, ruft das Programm topologische Anpassungsregeln auf, die bestimmte topologische Beschränkungen aufheben. Falls eine Transformation existiert, die wichtige Gebäudeeigenschaften beibehält, beginnt das Programm wieder mit Schritt 1; ansonsten wird keine Lösung erreicht.

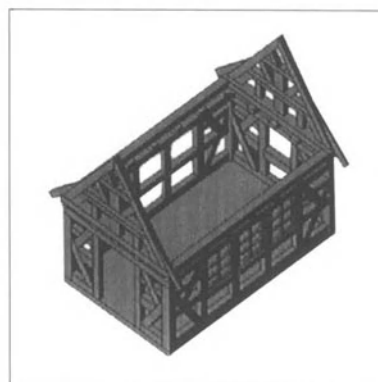
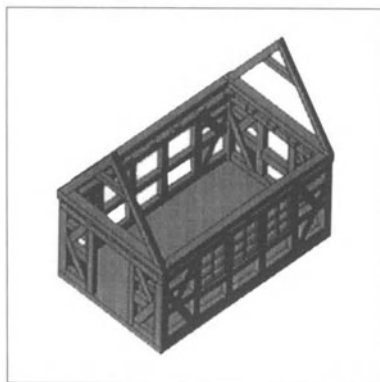
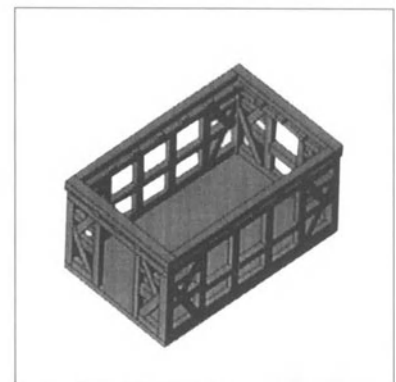
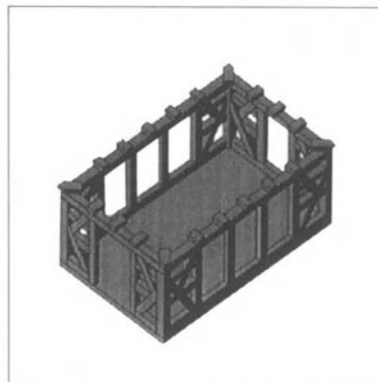
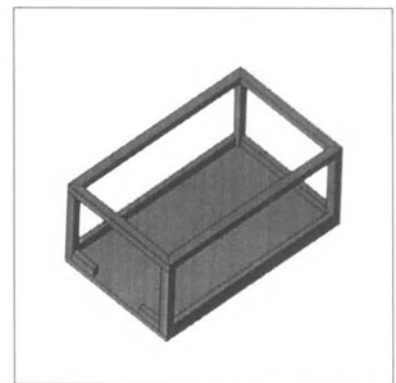
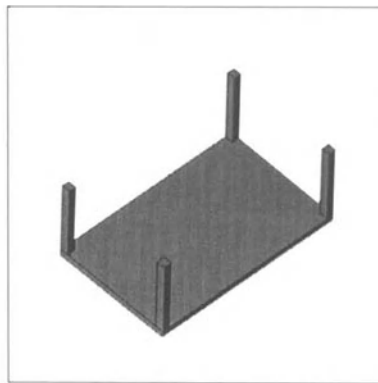
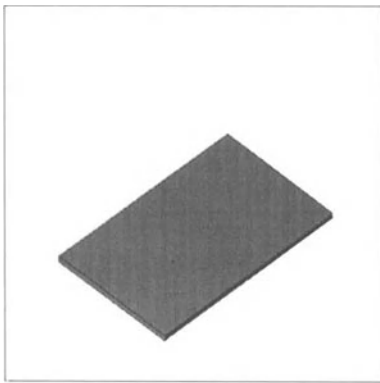
(8) Am Ende steht die visuelle Überprüfung des Ergebnisses, welche ein wichtiges Maß für den architektonischen Erfolg der Anpassung ist. Der Sinn der Simulation liegt darin, dem Betrachter ein interaktives Erle-

ben des adaptierten Entwurfs zu erlauben und Konflikte sofort zu erkennen.

Am Beispiel des Entwurfs und der Ausführung eines kleinen Gebäudes, das sich in nahezu idealer Weise für die Anwendung des fallbasierten Schließens eignete, lassen sich ausgewählte CAAD-Methoden und -Instrumente in der Praxis testen. Das Entwurfsproblem war die Erstellung eines vielseitig verwendbaren Raumes in unmittelbarer Nähe eines bestehenden Gebäudes. Den Kontext bilden ein Fachwerkhaus aus dem 17. Jahrhundert sowie Wald. Bestehende Eichenbalken aus dem 17. Jahrhundert in verschiedenen Längen und Erhaltungszuständen waren wiederzuverwenden. Natürlich ist die Wahl der Vorgaben bereits ein wichtiger Schritt für das Entwerfen und daher subjektiv. Vielleicht beeindruckt vom Wissen über die formalisierte Methode des fallbasierten Schließens, gestaltete sich der Entwurfsvorgang jedoch reibungslos. Nach grundsätzlichen Überlegungen folgten detailliertere Gedanken über das neue Gebäude, wobei die vorhandenen Eichenbalken als Konstruktionsmaterial eine wichtige Rolle spielten. In der Tabelle ist auf der linken Seite das traditionelle Denken beschrieben, auf der rechten Seite die entsprechende Anwendung des fallbasierten Schließens: Aus Zeitgründen war die Verzapfung aller Balken nicht machbar. Die maschinelle Bearbeitung war wegen des schlechten Zustandes der Balken (krumm, voller geschmiedeter Nägel, zum Teil bereits im Querschnitt erheblich durch frühere Verzapfungen geschwächt) ebenfalls nur beschränkt möglich. Die Adaptierung bestand darin, Verzapfungen mit minimaler Querschnittschwächung an kritischen Stellen vorzunehmen und die Konstruktion insgesamt durch Verschraubung zu stabilisieren.

Wir fanden schnell heraus, daß die normalerweise im fallbasierten Schließen wichtigen Tätigkeiten, wie

Die Stadien des Bauprozesses. Auch für diesen Vorgang diente die Herstellung des benachbarten Gebäudes als Fall.



Vergleich von Aspekten traditionellen und fallbasierten Entwerfens anhand eines Beispiels

Aspekte traditionellen Entwerfens

Material. Mitverursacht durch die verwendeten Materialien (Eichenholz für das Fachwerk, verputzte Füllungen und schiefergedeckte Dächer) strahlt das bestehende Fachwerkhaus eine nachahmenswerte Atmosphäre aus.

Gebäudetyp. Das neue Gebäude soll sich dem bestehenden Fachwerkhaus eindeutig unterordnen, es soll als unterstützender, nicht aufdringlicher Zusatz die Wirkung des Bestehenden verstärken.

Gebäudeform. Ein Anbau scheidet aus. In Frage kommt deshalb eine kleine Scheune in der Form der 'Primitive Hut' [Mitchell 1992a], dem Urtypus des Hauses. In Frage steht die Richtung des Daches.

Proportionen. Frühere Studien des Gebäudes förderten den Goldenen Schnitt sowie die Verhältnisse $1:\sqrt{2}$ und $1:1$ als wichtigste Proportionen zutage [Schmitt 1988a, S. 57]. Auch sie tragen zu der angenehmen Atmosphäre bei, die das Haus auf die Betrachter ausübt. Es bestand kein Grund, diese Proportionen im neuen Gebäude nicht zu verwenden.

Größe. Die bestehenden Balken wurden in einer Datenbank nach Länge, Breite, Höhe und Zustand erfasst. Daraus ergab sich die Möglichkeit, sie in ein CAD-Modell zu integrieren und einige erste Proportionsstudien anzufertigen.

Konstruktion. Das bestehende Haus ist mit verzapften Eichenbalken errichtet. Die Zwischenräume waren ursprünglich mit Lehm und Eichenstäben ausgefüllt.

Aspekte fallbasierten Entwerfens

Materialwahl bei Fachwerkhäusern und bei der Renovierung historischer Bauten. Materialien in der Fall-Datenbank für Fachwerkhäuser werden übernommen.

Typische Anordnung kleiner Scheunen bei Hofgebäuden dieses Typs. Diese unterstreichen die Dominanz des Hauptgebäudes und machen ihm keine Konkurrenz. Fall-Datenbank vorhanden.

Fälle sind ein First parallel zum Hausfirst und First senkrecht zum Hausfirst. Gewählt wird der First senkrecht zum Hausfirst, da so auf der Süd- und auf der Ostseite des neuen Gebäudes zwei positive Räume entstehen.

Gebäude der gleichen Art in derselben Gegend weisen erstaunliche Ähnlichkeiten in den Proportionen der Gebäudemodule auf. Deshalb liegt eine Weiterverwendung der Proportionen $1:1$, $1:\sqrt{2}$ und des Goldenen Schnitts nahe, die alle in der Fall-Datenbank vorhanden sind.

Die Fall-Datenbank enthält Fachwerkscheunen, die aber meist größer sind. In diesem Fall setzen die vorhandenen Balken den Maßstab.

Nach den vorangegangenen Entscheidungen liegt die Verwendung einer ähnlichen Konstruktionsart wie beim bestehenden Gebäude nahe.

Konstruktionsdetail mit dem portablen Computer als einzigem Entwurfs- und Dokumentationsinstrument.



Ansicht des Gebäudes von Süden in einer fortgeschrittenen Bauphase



Indexierung und Datenbankzugriff, beim fallbasierten Entwerfen weit hinter die Wichtigkeit guter Anpassungsmechanismen (Case Adaptation) zurücktreten. Als schwierig stellte sich die topologische Anpassung eines bestehenden Falles an eine neue Situation heraus, ohne dessen gute Qualitäten zu verlieren. Am vielversprechendsten ist dabei die Verwendung von Grammatik [Flemming 1988b, Shih 1993]. Damit stellen wir sicher, daß Räume nicht wahllos in Gebäuden herumgeschoben, sondern sensitiv der neuen Situation angepaßt werden. Dabei gibt es die Möglichkeit des additiven und des dividierenden Vorgehens. Für beide finden sich interessante Parallelen bei Richard Weiss [Weiss 1973, S. 157]. Er schreibt:

"Entwicklungsgeschichtlich betrachtet, entsteht der addierende alpine Typ aus dem Aneinanderfügen von Einzweckbauten zu einem nachträglichen Ganzen, das als solches die Teile, aus denen es addiert wurde,

immer noch klar erkennen läßt. Der dividierende Typ des nichtalpinen Gebietes erscheint im Gegensatz dazu als ein ursprüngliches Ganzes, das durch innere Aufteilung im Sinn eines Mehrzweckhauses den verschiedenen Wohn- und Wirtschaftsbedürfnissen dienstbar gemacht worden ist. Beim Mittellandhaus ist das Ganze des Wohnteils bestimmend, beim alpinen sind es die Teile. Der dividierende Grundriß wirkt unübersichtlich und komplex, der addierende übersichtlich und klar".

Als Kurzschluß könnte daraus folgern, daß Flemmings Methode der Zerlegung von Rechtecken sich für die dividierenden, Formengrammatiken dagegen für die addierenden Grundrisse eignet [Flemming 1986].

In Zukunft könnte sich fallbasiertes Entwerfen als eine Alternative zum automatischen Maschinenlernen entwickeln, bei dem ein Programm neue Regeln durch Verallgemeinern von Beispielen erlernt. Im fallbasier-

Blick von innen nach Süden. Die Dachbalken sind neu, auch dies eine gängige Praxis in Gebäuden dieser Art



Das benachbarte Wohnhaus als ursprünglicher Fall (rechts) mit dem Resultat der Adaptation.



ten Entwerfen geschieht die Verallgemeinerung nur zu dem Zeitpunkt, in dem das Wissen tatsächlich benötigt wird. Damit werden zwei Hauptprobleme des Maschinenlernens umgangen: Erstens ist oft nicht klar, welche Regeln gelernt werden sollen, und zweitens, in welchem Grad diese Regeln generalisiert werden müssen. Das letzte Problem tritt allerdings auch im fallbasierten Entwerfen auf, indem nämlich entschieden werden muß, ob ein bestimmter Fall für das jeweils aktuelle Problem anwendbar ist oder nicht. Wegen des paarweise erfolgenden Vergleichs der Beispiele ist dies jedoch wesentlich einfacher als beim automatischen Maschinenlernen.

Um einen bestehenden Fall auf ein neues Problem anzuwenden, ist es notwendig, die Problemstrukturen bestehender und zu untersuchender Fälle zu vergleichen. Bereits implementierte Case-Based-Reasoning-Programme erreichen dies dadurch, daß sie zunächst

eine kausale Analyse eines Falles in der Fall-Datenbank in bezug auf das aktuelle Problem erstellen und danach Aspekte dieser kausalen Problemstruktur mit dem aktuellen Problem vergleichen. Dabei wird weiterhin gebietspezifisches Wissen benötigt, das normalerweise durch ein traditionelles regelbasiertes System zur Verfügung gestellt wird. Jedoch ist die Zusammenstellung des Wissens für dieses System weitaus einfacher als im Fall eines vollständig generativen Systems, denn verlangt wird hier lediglich eine Analyse und keine Synthese. Außerdem muß dieses Wissen nicht komplett sein: Eine fehlende Analyse führt zum Auslassen eines möglichen Vergleichs; jedoch die Problematik eines solchen Falles ist vernachlässigbar, wenn die Fall-Datenbank genügend umfangreich ist. Daraus ergibt sich die äußerst attraktive Eigenschaft des fallbasierten Entwerfens, mit unvollständigen Daten arbeiten zu können.

Südfassade des neuen Gebäudes mit simulierten Füllungen der Fächer. Dieser Schritt ist wichtig für die Entscheidung über die endgültige Farbgebung.
Van der Mark, 1993



Intelligente Objekte

Zahlreich und treffend sind die Argumente, die man gegen Computer in der Architektur vorbringen kann. Für jeden neuen Vorschlag findet sich ein warnendes Beispiel, wohin diese Entwicklung führen könnte. Besonders deutlich wird dies auf dem Gebiet der Künstlichen Intelligenz. Bei der erstmaligen Erwähnung dieses Wortes kommt fast zwangsläufig die Feststellung, man solle erst einmal die natürliche Intelligenz unterstützen, das Ganze sei nicht ausgereift, und überhaupt habe man schlechte Erfahrungen mit diesem Humbug gemacht.

Intelligentes Objekt könnte sich zum neuen Reizwort entwickeln. Doch verbirgt sich dahinter eine sehr handfeste und langfristig für die Architektur nützliche Idee. Intelligente Objekte sind ein nächster logischer Schritt in der Entwicklung besserer Entwurfsumgebungen. Beispiele sollen die Idee der intelligenten Objekte erklären. Zu berücksichtigen ist dabei die Beobachtung, daß das, was heute noch als Semantik erkannt, morgen als Syntax angesehen wird.

Am Anfang standen Punkt und Linie, die sich durch Definition ihrer Koordinaten auf den Bildschirm projizieren ließen. In dem Moment, in dem diese Elemente als Objekte gesehen wurden, die die Anwender interaktiv bewegen konnten, war der erste Schritt in Richtung intelligente Objekte getan. Denn die Bewegung der Objekte bedingt die Definition und das Einhalten von Constraints, über die eine Einigung notwendig ist. Eine dieser Übereinkünfte ist, daß sich die Objekte in einer Ebene bewegen. Eine zweite Übereinkunft ist, daß ein Angreifen und Verschieben eines Endpunktes eine andere Auswirkung als das Angreifen und die Verschiebung des Mittelpunktes hat. Eine dritte besteht darin, daß das Objekt durch einfache Tastenkombinationen oder durch Verschieben in ein Mülleimer-Symbol in einen temporären

Puffer abgelegt oder völlig gelöscht werden kann. Diese 'primitiven' Übereinkünfte charakterisieren zum einen die Benutzeroberfläche und werden von Konsistenzgedanken aus dieser Richtung erforderlich. Sie lassen andererseits die Objekte in gewisser Weise intelligent erscheinen. Denn wie sonst kann die Linie 'wissen', daß man sie im ersten Fall insgesamt bewegen, im zweiten Fall aber nur einen Endpunkt verschieben möchte?

Dieses Problem ist auf zwei Arten lösbar, die ihre Entsprechung in anderen Bereichen der Wissenschaft und sogar der Gesellschaft haben. Überspitzt ausgedrückt, liegt die Wahl zwischen autoritär oder demokratisch, global oder lokal. Ein gutes CAAD-System wird Eigenschaften beider grundsätzlichen Möglichkeiten beinhalten. Die globale Lösung wäre, daß eine Hauptschleife ständig die Aktionen der Anwender überprüft. Wenn der Anwender eine Linie anwählt, trifft das Programm die entsprechenden Verzweigungen, und die für die Linie geltenden Constraints treten in Kraft. In der lokalen Lösung sind die Linien als individuelle Objekte repräsentiert und 'wissen', falls der Anwender sie anwählt oder aktiviert, was mit ihnen geschehen darf und was nicht. Dieses Wissen ist in Form von Methoden implementiert, die mit den Objekten gespeichert sind. Individuell sind diese Methoden abwandelbar, ein großer Vorteil gegenüber dem eher starren Schema der globalen Lösung.

Diese Qualität der lokalen Lösung erlaubt komfortable Konstruktionsvereinfachungen, die bei der praktischen Anwendung von großer Bedeutung sind. Ein Beispiel ist die geometrische Sensibilisierung der Objekte. In allen CAD-Programmen ist man in der Lage, den Endpunkt oder den Mittelpunkt eines Objektes direkt zu 'fangen'. Das bedeutet, daß beim Konstruieren der Endpunkt nicht mehr mühsam gra-

phisch gesucht oder numerisch definiert werden muß, sondern daß die Aufforderung an das Programm, den Endpunkt einer Linie zu suchen, und ein Klick in die entsprechende Region des Bildschirms den Endpunkt findet und mathematisch eindeutig definiert.

Eine weitere Steigerung des Konstruktions-Komforts ist mit der objektorientierten Implementierung einfach: Bewegt man den Cursor lediglich in die Nähe eines Objektes, zum Beispiel einer Linie, kann diese sogleich interessante Regionen bezeichnen, also Endpunkt und Mittelpunkt. Mehr noch, Schnittpunkte von Linien, Verschneidungen der Verlängerung von Linien und Kombinationen dieser Operationen, die auch beim traditionellen CAD noch große Schwierigkeiten bereiten, werden dadurch mit Leichtigkeit möglich.

Auf diese Art haben kommerzielle CAD-Programme bereits einen Großteil der ehemals unangenehmen und fehlergeplagten Konstruktionsarbeit erleichtert. Voraussetzung war die gemeinsame Basis von Geometrie und Algebra und die Eindeutigkeit der Beziehungen der Objekte zueinander. Obwohl einige mit CAAD Unerfahrene diese Möglichkeiten der 'intelligenten Cursors' als intelligentes Verhalten akzeptieren würden, ist dies heute schon ein als selbstverständlich vorauszusetzendes Merkmal moderner CAD-Programme geworden: vielleicht intelligent, aber sicherlich nicht mehr neu.

Die nächste Stufe der Intelligenz war die viel zitierte Fähigkeit von CAD-Programmen, auch das Fenster und die Tür zu bewegen, wenn die Wand verschoben wurde. In den meisten Fällen bedingte dies zunächst die Definition der Tür oder des Fensters als Teil der Wand. Einmal dem System mitgeteilt, konnte dies dann beide als Objekte verwalten und entsprechend

verschieben. Ein wirklich intelligentes Wandobjekt 'wüßte' von selbst, wenn ein Fenster eingesetzt wird, dies als Teil zu betrachten. Doch die Schwierigkeiten enden nicht dort. Was geschieht, wenn die Wand in ihrer Länge gedehnt oder gar gekrümmt wird? Wie verhalten sich in diesem Fall das Fenster oder die Tür? Im Grunde bedarf es dazu lediglich der Definition neuer Constraints für das Objekt. Doch im Gegensatz zu den anfangs erwähnten allgemeingültigen Konventionen sind bei komplexen, frei kombinierbaren Objekten die erlaubten und nicht erlaubten Aktionen nicht mehr eindeutig zu identifizieren. Die Kombinationen möglicher Aktionen wachsen schnell an, und die Gefahr endloser und zyklischer Abwägungen wächst.

An diesem Punkt angelangt, wird jedes Objekt zum mehr oder weniger komplexen Programm, das mit anderen Objekten oder Programmen interagiert. Und jedes Objekt muß nicht nur etwas über sich selbst 'wissen', sondern auch über seine Umwelt: Es ist in der Lage, Entscheidungen zu übernehmen. Nehmen wir die einfache Wand als Beispiel. Sie muß wissen, ob sie frei steht oder mit einer anderen Wand im Winkel verbunden ist, ob sie tragend ist, aus welchem Material sie besteht, ob sie innen oder außen liegt, ob sie Energie speichern soll und ähnliches mehr. Es mag seltsam klingen, aber wir müssen den Objekten, mit denen wir entwerfen, mehr Wissen und damit mehr 'Eigenverantwortung' geben. In der zunehmend komplexer werdenden Welt des Planens und Bauens sollte es die Möglichkeit geben, daß die Einzelteile sich selbst organisieren oder zumindest ihre wichtigsten Eigenschaften berechnen können, nachdem wir sie prinzipiell beschrieben und plaziert haben. Der Unterschied zum intelligenten Cursor ist lediglich quantitativ, nicht qualitativ.

Virtuelle Realität in der Architektur

Beginn der Exploration des virtuellen Raums. Ein Modell aus der Datenbank wird durch neue Oberflächen zum architektonischen Informationsträger umkodiert. Links: Objekt, rechts: Raum, Wenz, 1993

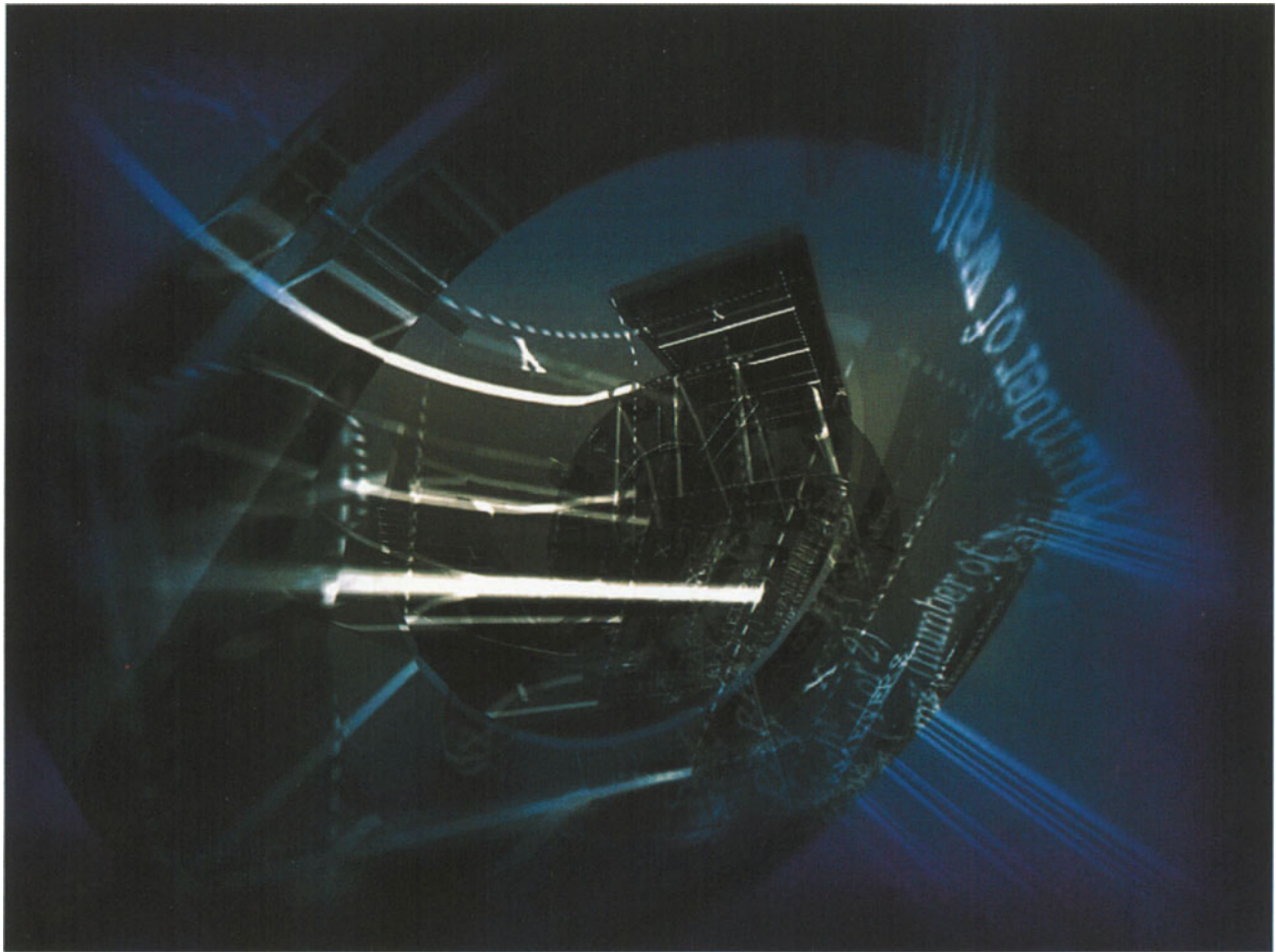


Mit dem Erscheinen der ersten Architekturzeichnung begann in der Baukunst eine Entwicklung, deren nächster konsequenter Schritt die Anwendung der Technik der Virtuellen Realität (VR) sein wird. Heute erlauben es die neuesten gedanklichen und technischen Möglichkeiten, dem Ziel einer virtuellen Architektur näherzukommen, die vor ihrer Ausführung in möglichst vielen ihrer Konsequenzen bekannt und erfahrbar ist. Die Komponenten der VR sind bekannt, neu ist ihre Kombination.

Die Virtuelle Realität basiert auf einem Modell der Wirklichkeit, das in vereinfachter Form im Computer als Datensatz vorhanden ist und mit dem die Betrachter interagieren. Dieses Modell macht Aspekte der Realität zugänglich, die außer dem Auge auch die anderen Sinne des Menschen ansprechen. Der Vorgang ist als Simulation bekannt, mit der in der Vergangenheit zum Beispiel die Geometrie, das Energieverhalten, das statische System oder die Kosten eines Projekts berechnet wurden. Neu an der VR ist, daß

das Datenmodell jetzt interaktiv zu explorieren ist und daß die Kombination der Simulationen eine Fülle neuer Eindrücke erlaubt. Der Übergang von der traditionellen Simulation zur VR ist fließend. Die beiden wichtigsten Charakteristika der VR sind die Interaktion und die Immersion, das bedeutet, die Objekte direkt manipulieren zu können und das Gefühl zu haben, von einem virtuellen Raum vollkommen umschlossen zu sein. Erreicht wird dies entweder mit VR-Helmen (Head Mounted Displays, HMD) oder durch Großprojektion in Stereo.

Die Architektur ist ein natürliches Anwendungsgebiet der VR. Jeder Plan, jede Perspektive versucht bei den Betrachtern eine Illusion zu erzeugen, die mit möglichst einfachen Mitteln eine möglichst vollständige architektonische Aussage macht. Allerdings verstehen die wenigsten Laien die Sprache der zwei- und dreidimensionalen Abstraktion genügend, um auf dieser Basis neue Projekte fundiert beurteilen zu können. Noch weniger werden dadurch die Zusammenhänge



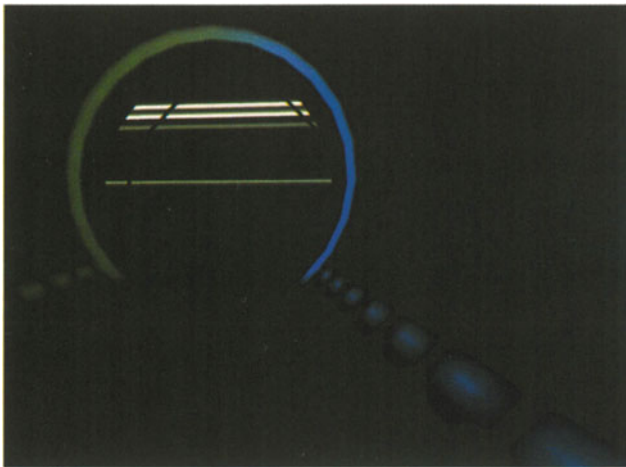
zwischen Form, Funktion, Verhalten und Kosten genügend klargestellt und abschätzbar. Ein virtuelles Modell mit hohem Realitätsgrad, das all diese Aspekte in integrierter Form berücksichtigt und das die Betrachter in jeder beliebigen Art erkunden können, wird eine große Hilfe darstellen.

Die neuen Möglichkeiten der VR verstärken den Bedarf nach einem sinnvollen und integrierten Modell für das Simulieren von Architektur - vom Entwurf bis zum Facility Management. Die Einführung von CAAD hat gezeigt, daß die geometrisch-graphische Repräsentation nur eine Form der Darstellung ist. VR wird die Vermutung bestätigen, daß die Beschränkung des Computers auf seine Funktion als elektronischer Zeichenstift eine Sackgasse ist. Denn damit werden lediglich die Fehler der Vergangenheit schneller gemacht. Vielmehr muß für die Idee des

Gebäudes als eines Organismus, in dem die Teile sinnvoll zusammenwirken, das richtige Modell gefunden werden.

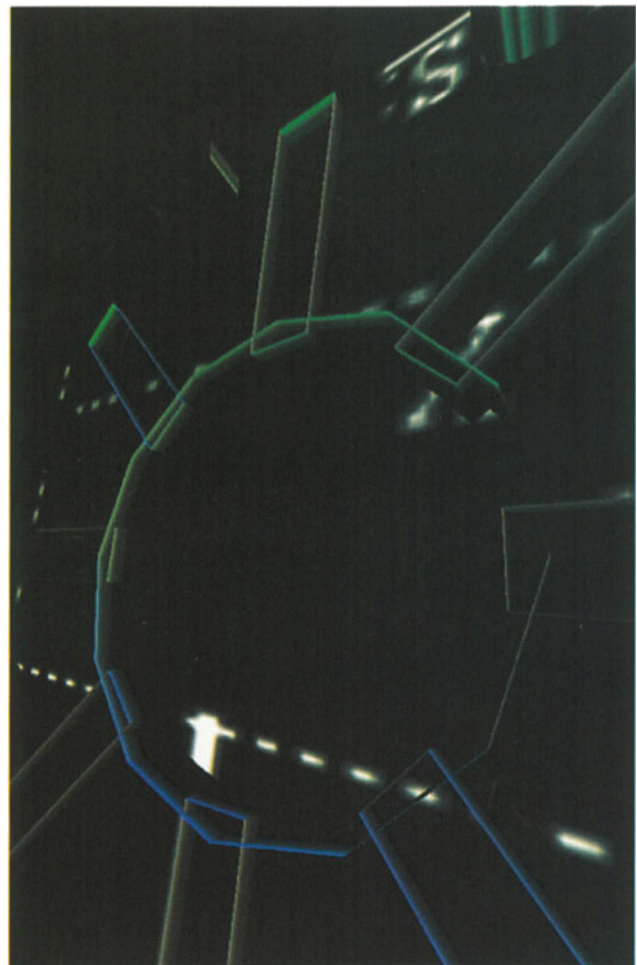
In diese Richtung weist die Arbeit der Gruppe Art+Com unter Edouard Bannwart in Berlin. Das Calibre Institut der Technischen Universität Eindhoven demonstrierte seine architekturbezogene VR-Installation bereits an der Konferenz CAAD futures '91 [Schmitt 1992b]. Mitarbeiter von Daniel Thalmann an der Ecole Polytechnique de Lausanne arbeiten ebenfalls auf diesem Gebiet. In den USA war die ACADIA Konferenz 1991 diesem Thema gewidmet. Die Fraunhofergesellschaft in Stuttgart unterhält ein VR Demonstrationszentrum.

Die Herstellung einer VR-Umgebung für die Architektur setzt extrem schnelle Hardware und intelligente Software voraus. An der Architekturabteilung der ETH



Oben und rechts: Dreidimensionale Benutzerschnittstellen ermöglichen als gateways die Orientierung im Datenraum und regeln Zugriffsberechtigungen.

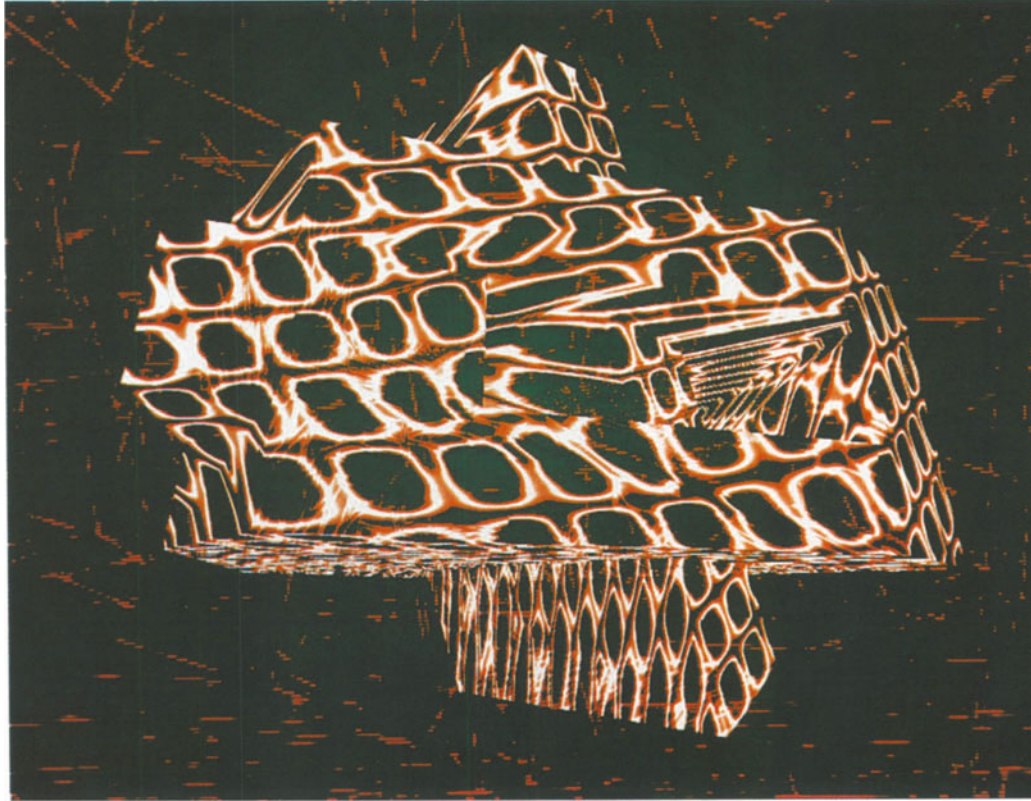
Links: Eintritt in den virtuellen Raum unter beschleunigter räumlicher Wahrnehmung. Perspektivische Raumerkennung und digitale Materialeigenschaften folgen neuen Gesetzen. Wenz, 1993



Zürich besteht daher für Forschung und Lehre ein Architectural Space Laboratory, um die Möglichkeiten der VR in der Architekturlehre auszuloten und vielversprechende Richtungen für Forschung und Praxis zu definieren. Das ASL erlaubt für Forschungs- und Lehrzwecke die Simulation neuer Entwurfs-, Planungs- und Ausführungsvorgänge, die bisher in Kombination nicht möglich waren. Der Rahmen dafür ist das Architectural Space Laboratory (ASL), in dem eine sensorische Architektursimulation in visueller, akustischer und haptischer Hinsicht angestrebt wird. Das ASL testet Szenarien für das Architekturbüro des nächsten Jahrhunderts, in dem Bauherrschaft und Architekt die verschiedenen Aspekte eines neuen Gebäudes gemeinsam erleben. Das ASL ist kein isoliertes Labor. Schon jetzt sind die Computer des ASL an weltweite Netzwerke angeschlossen. Die Vernet-

zung ermöglicht ein hohes Maß an Kommunikation sowie die interaktive Simulation von Entwürfen oder Gebäuden sogar in anderen Ländern und Kontinenten.

Uns ist es wichtig, daß VR in der Architektur nicht eine Technik für das Individuum wird, sondern daß sich mehrere Beteiligte in der virtuellen Architektur bewegen und damit interagieren können. Spezielle, von uns entwickelte Software erlaubt es, intelligente Objekte, aus denen die Modelle aufgebaut sind, direkt zu manipulieren. Diese intelligenten Objekte können physische Äquivalente besitzen, wie Gebäudeelemente oder Möbel, oder sie können funktionalen Charakter haben. Es bietet sich dadurch erneut die Möglichkeit des partizipatorischen Entwurfs, in dem sich die an Planung und Bau Beteiligten und die Bauherrschaft gleichermaßen bewegen.



Materialität von virtuellen Objekten kann durch prozedural bestimmte Texturen vermittelt werden. Im Beispiel verunklaren diese die exakte dreidimensionale Form und schaffen so eine zusätzlichen Wahrnehmungsebene. Wenz, 1993

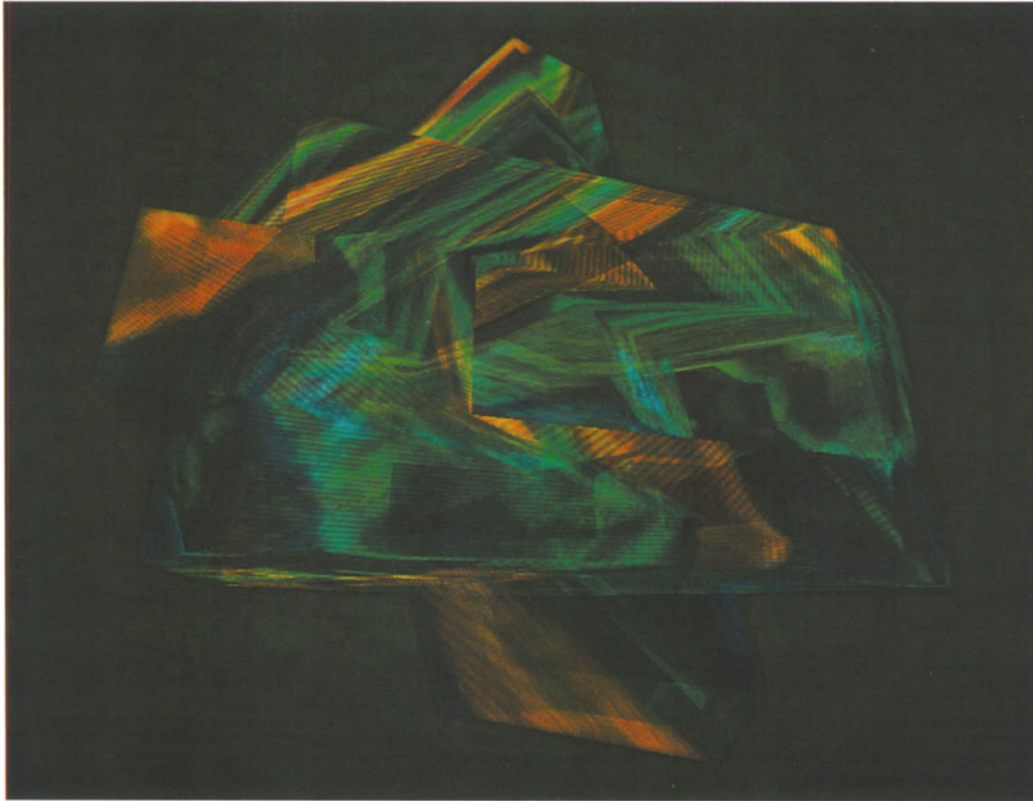
Damit wird die VR-Umgebung nicht nur für Präsentations-, sondern auch für Entwurfsaufgaben nutzbar. Mit den intelligenten Objekten gekoppelte Energie-, Kosten- und sonstige Informationen, die normalerweise nicht sichtbar oder direkt erfahrbar sind, werden im ASL simuliert. So wird es möglich sein, sich im Modell eines Konzertsaals zu bewegen und an jeder Stelle über das Audio-System die entsprechende Akustik zu überprüfen. Im Modell eines Gebäudes wird in den einzelnen Räumen der Temperaturverlauf über das gesamte Jahr sichtbar gemacht und über die Manipulation der Fenster oder Wände auf die gewünschte Bandbreite gebracht werden.

Es existiert bereits eine Warteliste von Anwendungen, die von der neuen Technik Gebrauch machen werden. Unter anderem sind dies sehr große Bauvorhaben mit möglichen schwerwiegenden Umwelteinflüssen, sowie die Ergebnisse von Forschungsaufträgen,

die im ASL realitäts- und praxisnäher beurteilt werden können.

Bei den wissenschaftlichen Applikationen des ASL haben wir das Ziel, durch Ausweitung der Wahrnehmungsfähigkeiten neue Zusammenhänge zu finden und zu erklären. Bekannt sind die Rekonstruktionen historischer Architektur, von exakten Zeichnungen und Aquarellen bis hin zu Computermodellen. Obwohl dabei meist nur die Form der nicht mehr vorhandenen Architektur zum Ausdruck kommt, ergänzen wir mit unserer Erinnerung weitere Merkmale vergangener Zeiten, wie wir sie aus Bildern oder Beschreibungen kennen. Diese Aufgabe kann eine VR-Umgebung besser erfüllen, indem sie akustische, farbliche oder atmosphärische Eindrücke ergänzt.

Wir haben diesen Weg mit der Rekonstruktion des römischen Avenicum beschritten. Doch konnten wir das Computermodell der Stadt bei weitem nicht aus-



Hier definiert ein Videogemälde des Künstlers Henning Timcke als Textur den emotionalen Ausdruck des Objekts. Dieselbe Form wie im linken Bild wird dadurch mit vollkommen neuen Inhalten assoziiert. Wenz, 1993

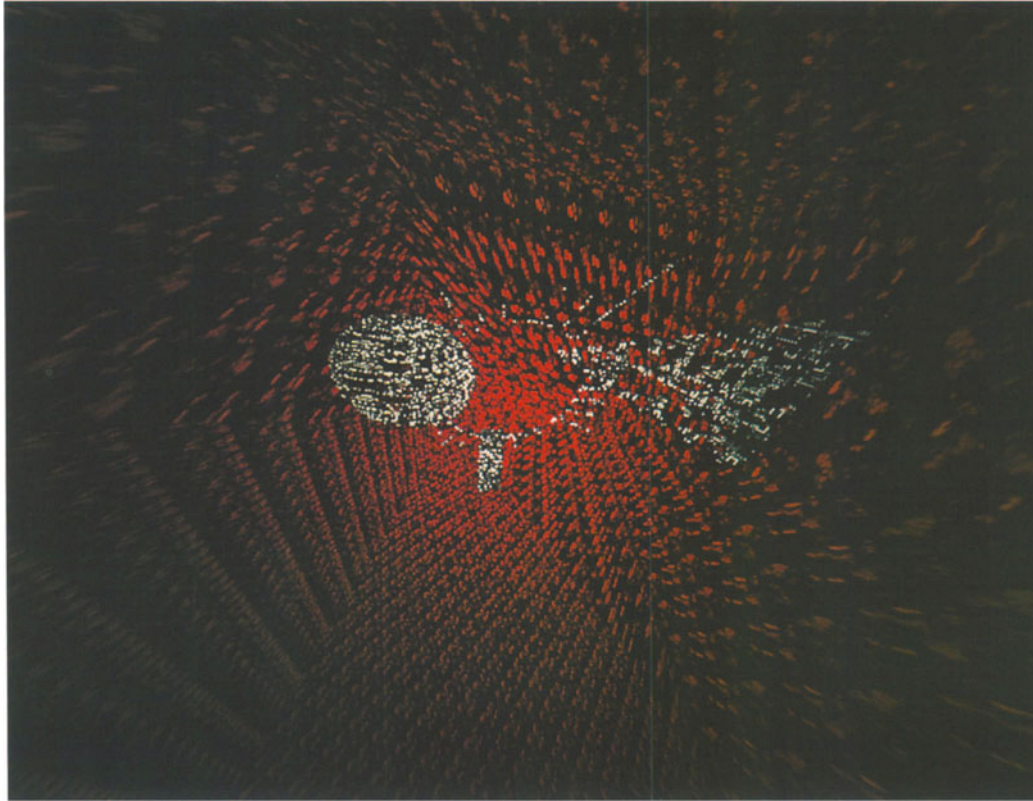
nutzen. Im ASL dagegen liefert ein virtuelles Modell von Avenicum zusätzliche Erfahrungen. Besucher werden dann die simulierte Stadt besuchen und sich interaktiv durch eine multimediale Welt der Antike bewegen. Forscher werden die Rekonstruktion mit den neuesten Ausgrabungen vergleichen und Korrekturen anbringen können.

Virtuelle Zukunft

Die Holzlattengerüste, die bei in der Schweiz geplanten Bauvorhaben die Umriss des neuen Objekts zeigen sollen, sind eine sehr abstrakte Simulation des neuen Gebäudes, die keinerlei Aussagen über Material und andere wichtige architektonische Merkmale macht. Bedenkt man, daß diese Gerüste als wichtige Entscheidungsgrundlage dienen, dann wird der

Bedarf nach einer realistischeren Simulation sofort klar. Auch traditionell präsentierte Projekte sind eine Art virtueller Architektur, denn sie versuchen, potentielle Bauherrschaften mit allen erlaubten Mitteln von der Qualität der Idee zu überzeugen. Besonders deutlich wird dies bei Wettbewerben.

Um große Bauvorhaben gibt es regelmäßig Auseinandersetzungen zwischen allen Beteiligten. In der weitgehend bebauten Umwelt der industrialisierten Länder ist jede Neuerung, jedes neue Gebäude ein Eingriff, der viele Anlieger direkt betrifft. Es ist daher absolut notwendig, die Konsequenzen dieser Eingriffe den Betroffenen vorher so klar wie möglich vor Augen zu führen. Dies gilt vor allem für die positiven Aspekte neuer Projekte. Die virtuelle Architektur bietet einen Ausweg aus der ungunstigen Situation, jede Änderung nur als Gefährdung zu betrachten. Architektinnen und Architekten werden VR dazu nutzen,



Ein Node eines globalen Netzwerks im Datenraum. Unterschiedliche geometrische Systeme repräsentieren entsprechende Organisationsstrukturen der Informationseinheiten.
Wenz, 1993

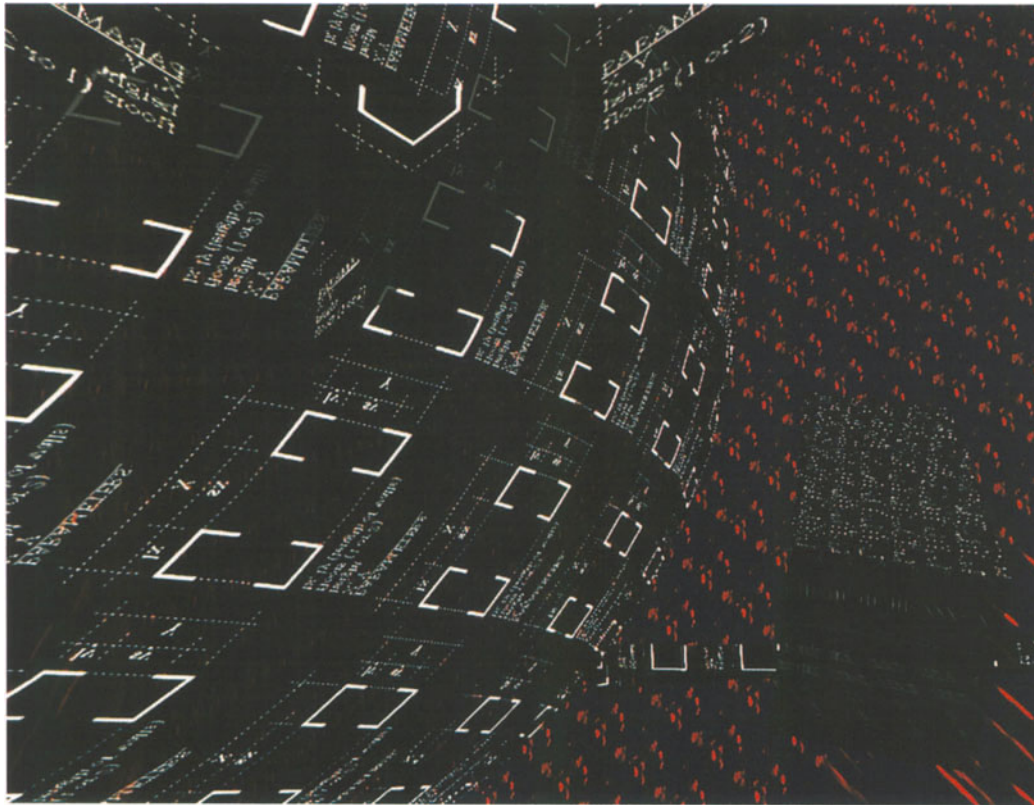
ihre Kompetenz bei der aktiven Gestaltung der Umwelt zu beweisen. VR wird natürlicherweise dabei helfen, mehr als nur die formal-geometrischen Aspekte der neuen Projekte zu vermitteln.

Blendung oder Aufklärung?

'Cogito, ergo sum - ich denke, also bin ich - Video, ergo est - ich sehe, also ist es' [Degler 1993, S. 151]. Die Virtuelle Realität bietet ihren Gegnern beliebig viele Angriffsflächen. Abgesehen davon, daß es sich hier um eine sehr kapitalintensive Technik handelt, liegt der Verdacht nahe, sie könne zur Blendung potentieller Entscheidungsträger benutzt werden. Die Tatsache, daß ähnliches bei der Einführung der Photographie befürchtet wurde - und auch eintrat -, ist ernstzunehmen. In der Tat bildet die VR heute die höchste Form der Sinnestäuschung, deren Beherrschung, wie in der Photographie, eine wahre Kunst

ist und die, wie diese, langfristig zum normalen Kommunikationsrepertoire der kreativen Berufe gehören wird. Der Entzug von "erlebbarer, fühlbarer Wirklichkeit" führe zu einer "einseitigen Beanspruchung der visuellen Wahrnehmungssinne und des Vorstellungsvermögens" und damit zur "Verwischung von Realität und Wiedergabe von Realität" [Degler 1993, S. 153]. Die Befürchtungen des Aachener Wissenschaftsethikers Matthias Gatzemeier sind in der Tat bereits bei vielen Konsumenten der neuen Technologie eingetreten. Doch was heute als beängstigende Möglichkeit zählt, ist morgen oft bereits verarbeitete Normalität, wie die Einführung von CAD in den Architekturbüros zeigt.

Jede neue Technologie hat dieses negative Potential. Durch die Einführung der VR an den Lehrstätten und Universitäten ist aber die Hoffnung gegeben, daß das Gefühl einer ethischen Verantwortung, welche die Beherrschung der neuen Technik notwendig macht, von Anfang an mitgebildet wird. Denn der



Dreidimensionale Strukturen dienen im interaktiven Netzwerk als Informationsträger und Benutzerschnittstellen. Links eine Random-Access Struktur, rechts ein Array. Wenz, 1993

Hauptzweck der VR ist die Aufklärung, also die möglichst wahrheitsgetreue Simulation eines neuen Zustandes mit den besten zur Verfügung stehenden Modellieretechniken.

Der Unterschied zwischen Realität und Simulation erreicht durch die Einführung von VR ein neues Minimum, zugleich nimmt der Abstraktionsgrad der architektonischen Präsentation weiter ab. Dies bedeutet, daß Laien eine größere Chance haben werden, ein neues Gebäude zu verstehen und zu beurteilen. Es bedeutet auch, daß der Freiraum und der Verantwortungsgrad der Architektur sich um eine zusätzliche Dimension erweitern.

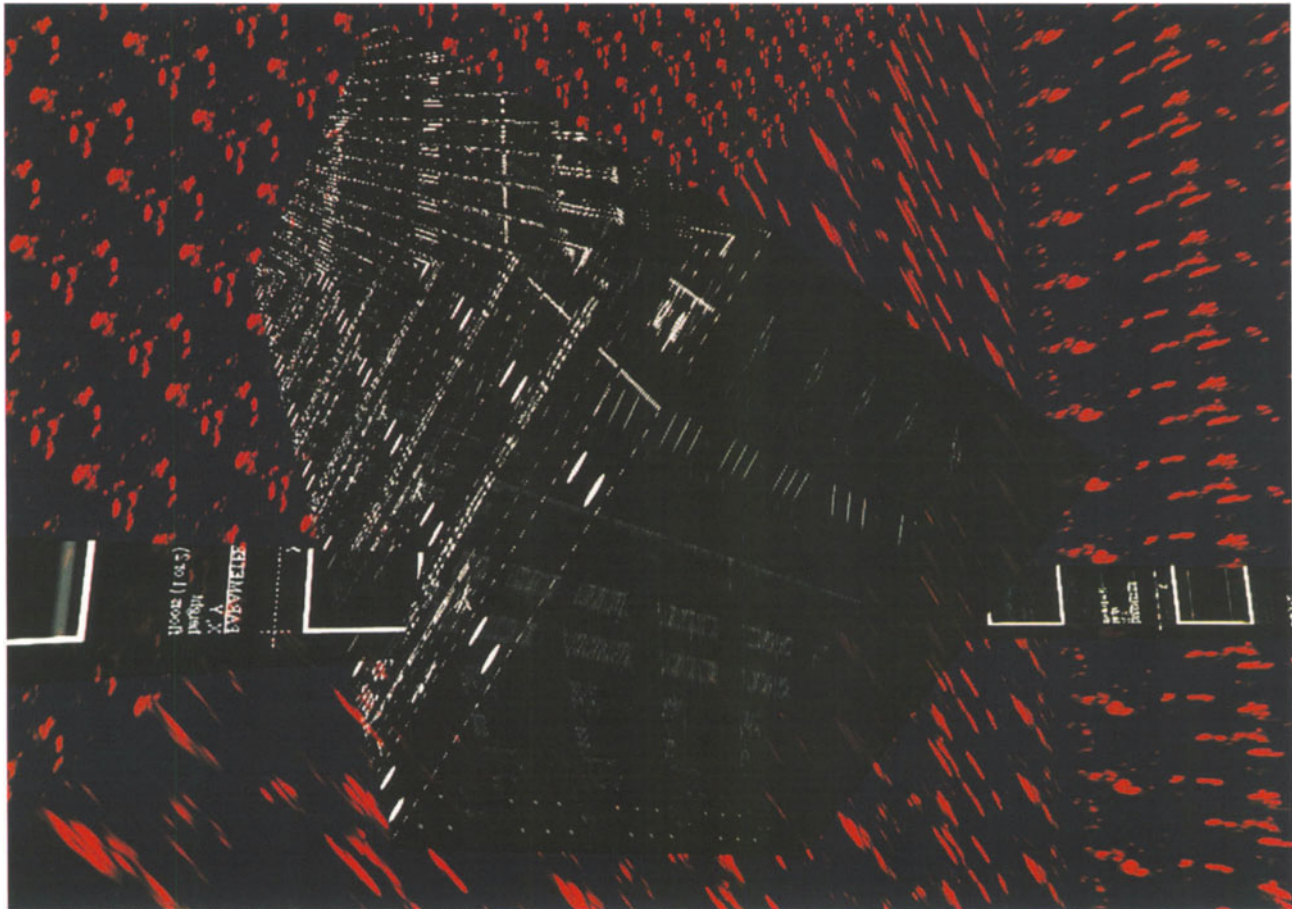
Kosten und Nutzen

Die Kosten für eine leistungsstarke VR-Umgebung für architektonische Anwendungen betragen ein Vielfaches einer Standard-Workstation. Dies schließt Hard-

ware, Software, audiovisuelle Ausrüstung und bauliche Maßnahmen ein. Damit ist klar, daß es kurzfristig nur wenige Büros mit dieser Ausrüstung geben wird. Trotzdem hat die Betrachtung eines Architekturprojekts mit dieser neuen Technologie größte Vorteile gegenüber traditionellen Zeichnungen oder Modellen.

Mit VR sind - anders als auf traditionellen Medien - die guten und schlechten Eigenschaften und der Gesamteindruck eines Gebäudes nicht nur integriert darstellbar, sondern auch zu manipulieren und sofort zu evaluieren. Hier sei der Vergleich zur Medizin erlaubt, in der Apparate in ähnlicher Preisklasse, sobald ein Bedürfnis dafür besteht, angeschafft und die Kosten an die Patienten weitergegeben werden. In der Architektur könnten sich solche Installationen bei größeren Projekten schnell bezahlt machen, wenn es gelingt, dadurch folgenschwere Fehler in der frühen Entwurfsphase zu vermeiden.

Die Fassade des Datenarrays zeigt vertikal gegliederte Informationsschichtungen.
Die Struktur ist mit weiteren Einheiten durch einen sequentiellen Strang verbunden. Wenz, 1993



Auswirkungen

VR wird den Architektur-Arbeitsplatz kostenintensiver machen. Die bereits in CAD getätigten Investitionen werden jedoch durch VR geschützt und erlauben erst die vollständigere Nutzung der Datensätze, in denen viele unerwartete Informationen schlummern. Denn drei- und mehrdimensionale Modelle sind die Voraussetzung für den erfolgreichen Einsatz von VR in der Architektur.

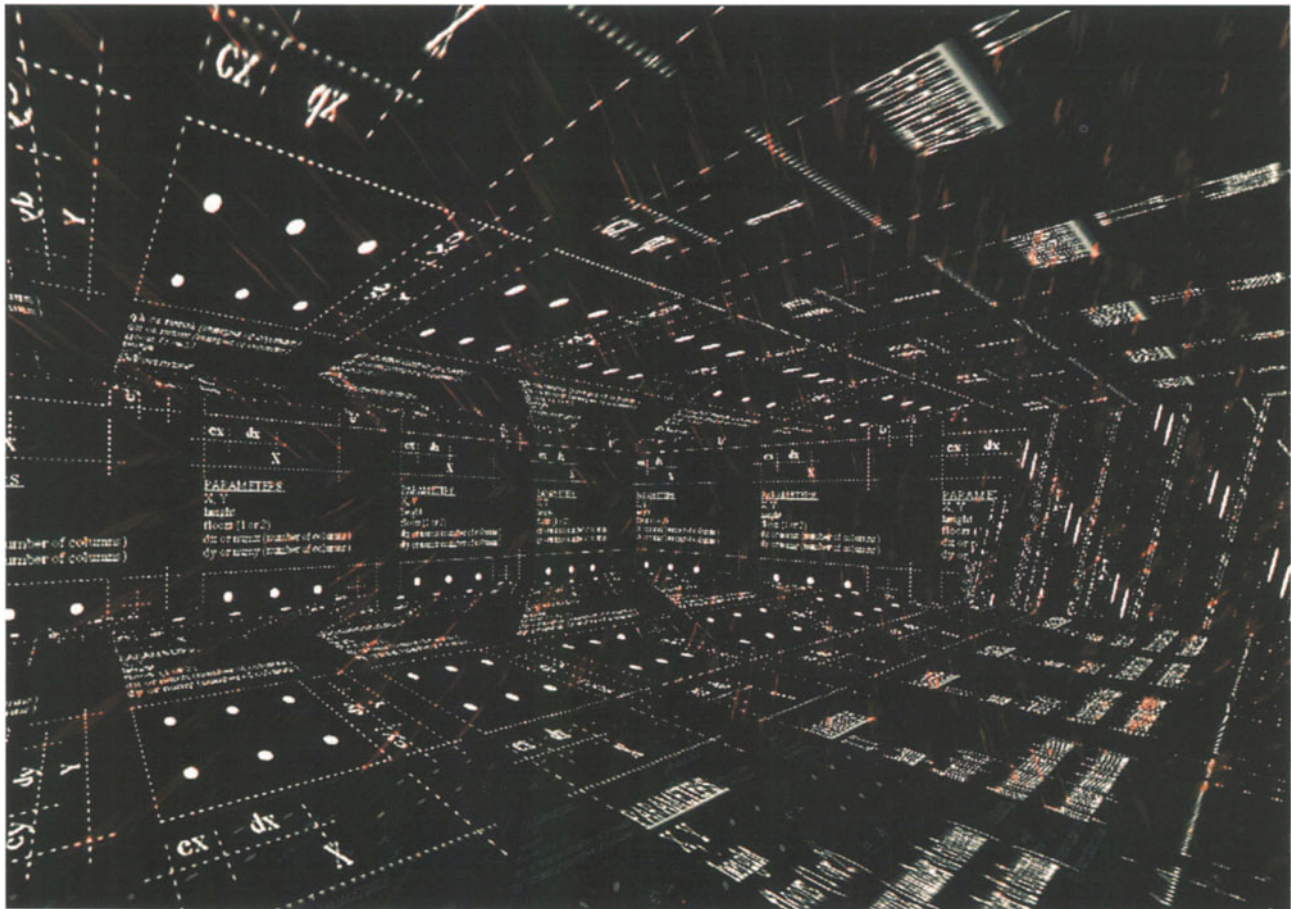
In der Architekturlehre wird in naher Zukunft nur ein kleiner Teil der Studentinnen und Studenten im Virtuellen Raum entwerfen können, alle werden jedoch die neue Technik erleben und testen können. Am Beginn wird das Darstellen historischer und zukünftiger Architektur mit dem neuen Mittel stehen, um so eine Bibliothek hervorragender Beispiele aufzubauen.

Darauf wird der Einsatz der VR im Entwurf selbst folgen.

Im Planungsprozeß wird VR nicht mehr wegzudenken sein. Von den vielen möglichen Anwendungen werden sich die Beurteilung durch von Neubaumaßnahmen Betroffene und durch die Fachleute am ehesten durchsetzen. So können Unklarheiten aufgeklärt oder Schwachpunkte entdeckt werden. Die vorausgehende Beurteilung eines Projekts im virtuellen Raum durch Fachleute sollte die Einhaltung aller entsprechenden Vorschriften sicherstellen und langwierige Rückfragen eliminieren.

Noch wichtiger aber kann sich die VR für die Planung in Entwicklungsländern erweisen, also für die Gebiete, in denen die Wohnungsnot am größten ist. So lassen sich die statischen Qualitäten verschiedener Bauweisen bei Katastrophen, wie Erdbeben, Stürmen oder

Innenansicht einer Array-Schicht aus der Benutzerperspektive.
 Alle begrenzenden Flächen sind als Informationsträger ausgebildet. Wenz, 1993



Überschwemmungen, testen oder die Materialbeständigkeit und -alterung simulieren. Auch für die Simulation der Verkehrsentwicklung, der Schadstoffverbreitung und des Städtewachstums bestehen Modelle, die sich in einer integrierten VR-Umgebung zu einem lebenswichtigen Instrument entwickeln könnten.

Eine virtuelle Architektur?

Es stellt sich natürlich die Frage, ob wirkliche Architektur überhaupt noch notwendig ist, wenn eine virtuelle Architektur Dinge erlaubt, die physisch schwer ausführbar sind. Ist es nicht vorstellbar, daß Bewohner eines Hauses sich jede erwünschte Wohn- oder Erlebnissituation über eine VR-Umgebung 'abrufen' und erzeugen werden? Besonders in Zeiten, da die

Abstraktheit des Lebens-, Wohn- und Arbeitsraumes ein Höchstmaß erreicht hat, ist es durchaus denkbar, daß modernste Technik zum Rückzug in ein virtuelles Biedermeier, Rokoko oder in die Römerzeit benutzt wird. Doch ist sicher anzunehmen, daß die größere Freiheit im Modellieren und in der integrierten Evaluation zu neuen Entdeckungen und architektonischen Erfindungen führen wird.

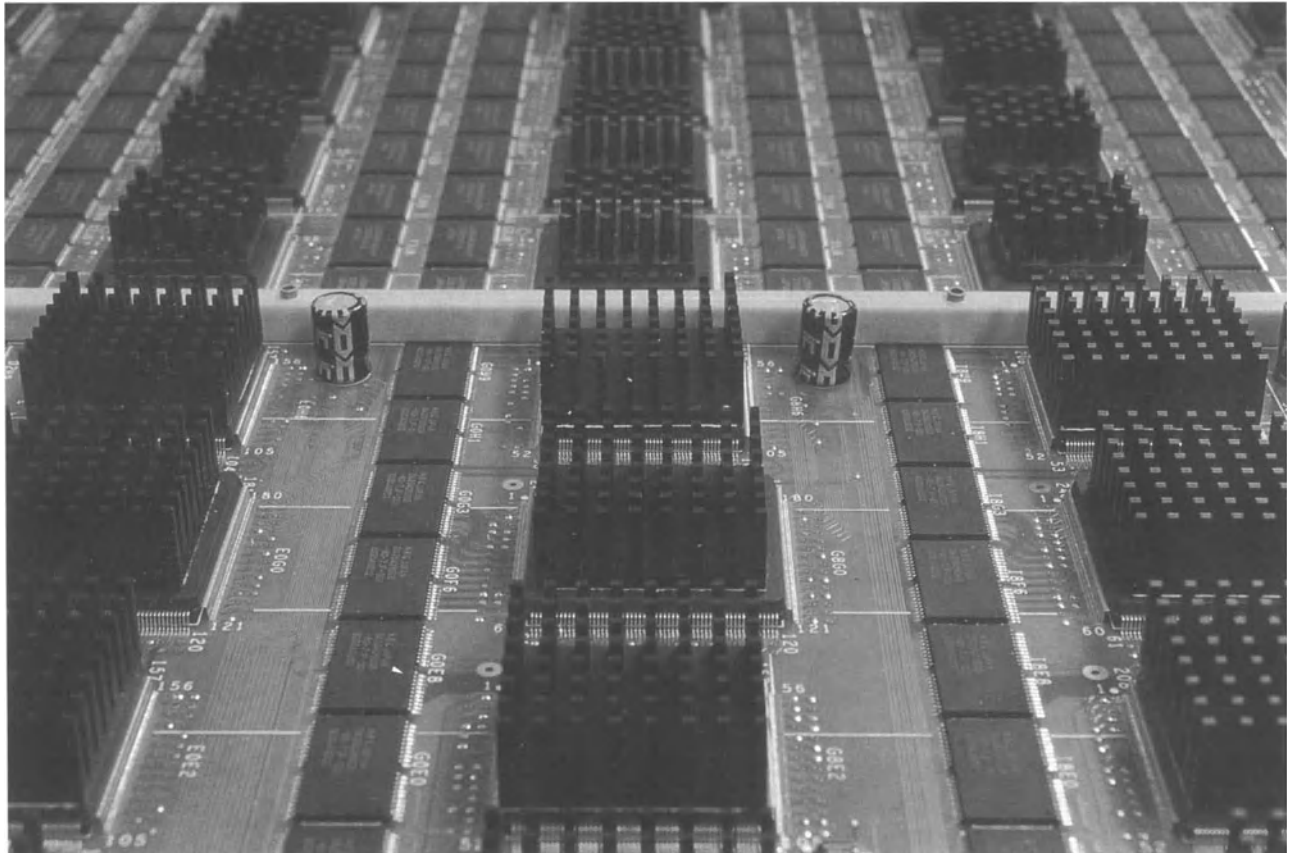
Meine Hoffnung ist, daß die virtuelle Architektur nicht zum Selbstzweck wird, sondern ein vorbereitender Schritt in Richtung auf eine verantwortungsvollere Architektur ist. Das bedeutet, daß mit Hilfe der Technik der Virtuellen Realität nicht sichtbare, aber essentielle Aspekte von Gebäuden direkt simulierbar werden, um so die Qualität der Entscheidungen zu verbessern - eine vielversprechende Herausforderung.

Machina

Einleitung

Machina - die Maschine

Blick in das Innere eines Computers - Memory und Graphikprozessoren einer Silicon Graphics Reality Engine



Machina - dieser Teil des Buches ist der Maschine und den Programmen gewidmet, die Computer zum Arbeiten benötigen. Das Kapitel machina beschreibt die Computerumgebung als Ganzes, kurz Computereinfrastruktur genannt. Zu Beginn steht eine allgemeine Schilderung der Entwicklung und der Rolle der Maschine in der post-industriellen Gesellschaft, sowie der Programme und ihrer Arbeitsweise. Es folgen Betrachtungen zu einigen architekturenspezifischen Aspekten des Computers und über die speziellen Anforderungen an die Maschine im Architekturbüro. Am Ende steht ein Ausblick mit Gedanken über die

Rolle der Künstlichen Intelligenz (KI) und die Bedeutung des Ausbaus der Netzwerke. Beschrieben werden im ersten Teil dieses Kapitels lediglich kommerzielle Computer und Programme, die bereits heute verfügbar sind. Im Gegensatz dazu stehen die in den ersten beiden Kapiteln behandelten innovativen Lehr- und Forschungsprogramme sowie Entwicklungen, die noch keine weite Verbreitung im kommerziellen Markt gefunden haben. Dieser letzte Teil des Buches ist daher für die Technologie- und Entwicklungsabschätzung der Praxis von Interesse. Er ist zeitabhängig und sollte entsprechend verstanden werden.

Die Maschine in der Gesellschaft: Materielle und intellektuelle Infrastruktur

“Die neuen Analphabeten sind erwachsen. Sie können lesen und schreiben, aber sie verstehen nicht die Sprachen der Computer” [Pardey 1992].

Es dauerte weniger als ein Jahrzehnt, bis der Computer in der industriellen Gesellschaft zum wichtigsten technischen Instrument wurde. Die achtziger Jahre des 20. Jahrhunderts, in denen der Dienstleistungssektor der größte Faktor der westlichen Volkswirtschaften wurde, sahen eine unaufhaltsame Verbreitung der Systeme und der damit einhergehenden Infrastruktur. Es ist nicht verwunderlich, daß erst mit der Einführung des ‘Personal Computer’ (PC) im Jahre 1981 die Massenverbreitung der neuen Technik einsetzte. Denn im Gegensatz zu den bis dahin üblichen Maschinen brauchten die neuen Computer lediglich eine Steckdose und einen Arbeitsplatz - und wurden so überall einsetzbar. Erst später erkannte man, daß mit isolierten Arbeitsplätzen wenig modernes Arbeiten möglich ist, woraufhin eine Welle von Peripherie- und Infrastrukturinvestitionen begann, die bis heute noch nicht abgeschlossen ist.

Der Computer ist die perfekte Projektionsmaschine: Wir projizieren eigene Gedanken und Verhaltensweisen auf die Maschine, und sie reproduziert sie, mehr oder weniger überzeugend, je nach Qualität des Programms. Immer muß dabei im Gedächtnis bleiben, daß die Maschine mit einem oder mehreren computerspezifischen Abstraktionsmodellen arbeitet, die sich wesentlich von unseren gängigen Abstraktionen unterscheiden, denn nur so lassen sich die besonderen Fähigkeiten der neuen Technologie ausnutzen. Noch sind alle Programme von Menschen geschrieben, die wissenschaftliche Vorgaben und eigene Vorstellungen einbringen. Jedes Programm trägt somit die persönliche Note der Programmierenden oder der

Programmentwickler-Gruppe. Besonders zutreffend ist diese Beobachtung in CAD-Programmen, wo die Arbeitsweise von Architektinnen und Architekten bisher nur in spezifischen Bereichen Unterstützung findet.

Im Gegensatz zu vielen anderen Technologien gelangen Computer verstärkt über die Gruppe der Jugendlichen und Kinder in die Gesellschaft. Dies führt zu interessanten Verschiebungen im Wissen. Haben in anderen Technologien die Erwachsenen oder besondere Gruppen ein Wissens- und Zugangsmonopol, so ist im Fall des Computers die Situation völlig anders. Das meiste Wissen besitzen die 20- bis 40-Jährigen, doch die größte Fingerfertigkeit und Neugier haben die fünf- bis 15-Jährigen. Doch während sich in der ersten Gruppe noch viele Programmierer finden, die das ‘Wesen des Computers’ zu ergründen suchen, sind die Jüngeren zum einen Teil Konsumenten, begeistert von den immer besser und raffinierter werdenden Software-Applikationen, zum anderen Teil aber auch Entdecker, welche die virtuelle Realität einer Programm-Umgebung zu explorieren suchen. Der von Erziehern bis vor kurzem gehegte Traum, in der neuen Wissenschaft der Computer die Ausbildung von Grund auf zu betreiben und jeden Schritt verständlich und nachvollziehbar zu machen, ist inzwischen von der Komplexität der Programme und der Vielzahl der Entwicklungsrichtungen zur Illusion geworden.

Die Maschine - Hardware

In der Hardware hat sich in den letzten Jahrzehnten eine interessante Entwicklung abgespielt. Am Anfang standen riesige, raumfüllende Rechenmaschinen,

technische Architekturen in sich selbst, mit noch verständlichen Relationen zwischen 'Rechenaufwand' und Rechnergröße. Diese energieintensiven Maschinen, ausgestattet mit elektrischen Röhren und bedient von Herren in weißen Kitteln, haben das frühe Bild des 'Computers' geprägt.

Schnell rückten die kleineren, mit Transistoren bestückten 'Mainframes' und 'Minicomputer' nach und verdrängten ihre Vorgänger. Die 'Reinheit' der Wissenschaft, die diese Maschinen und ihre Bediener verkörperten, ging einher mit der 'Reinheit' der Umgebung, in der sie standen. Die Architektur nahm sich dieser Notwendigkeit nur zögernd an, meist mußten die ersten Computer in fensterlosen Kellergeschossen mit Sicherheitsschleusen und Doppelböden untergebracht werden. Doch vereinzelt wurden sie auch zum visuellen Mittelpunkt des Geschehens, wie etwa an der University of Waterloo in Kanada.

Gerade, als diese Objekte das Interesse der Architektinnen und Architekten zu wecken begannen, da für ihre architektonische Unterbringung immer mehr Wissen notwendig wurde, entstand die nächste Maschinengeneration. Mit der Entwicklung der ersten CPU als integrierte Einchip-Schaltung begannen Mikrocomputer und Personal Computer seit Beginn der achtziger Jahre, sich in allen Bereichen durchzusetzen. Zu Beginn der neunziger Jahre schätzte man deren Zahl weltweit auf über 100 Millionen, mit weiter steigender Tendenz.

In der Geschwindigkeit und der Größenentwicklung der Rechner spielten sich faszinierende Entwicklungen ab. Seit Jahren verdoppelt sich die Rechenleistung fast jedes Jahr. Um die Rechner in ihrer Leistung untereinander vergleichen zu können, bedient man sich verschiedener Maßstäbe oder 'Benchmarks'.

Bekanntere, wenn auch nicht immer aussagekräftige Benchmarks sind MIPS (Million Instructions Per Second) und MFLOPS (Million Floating Point Operations Per Second).

Die Speicherkapazität der Rechner nimmt ebenfalls rapide zu. Man unterscheidet zwischen dem Hauptspeicher (RAM, Random Access Memory, vergleichbar dem Kurzzeitgedächtnis), und der Speicherkapazität der Festplatten (Hard Disk, vergleichbar dem Langzeitgedächtnis), der Disketten (Floppy Disk), der Wechselpplatten (Removable Harddisk), der optischen Datenträger (CD, Compact Disc) oder der Magnetbänder. Beeindruckend ist die Entwicklung bei den Magnetbändern, deren Kapazität sich in dem Jahrzehnt von 1983 bis 1993 von etwa 2 Megabytes auf etwa 5 Gigabytes erhöhte. Sie wahren weiterhin ihre Funktion als Massenspeicher und werden in großen Organisationen wegen der gigantischen Datenmengen von Kassettenrobotern geladen und verwaltet. Die Speicherung auf CD wird langsam populär; Baukataloge, Bedienungsanweisungen und ganze Betriebssysteme inklusive Soforthilfe (on-line Help) kommen so zur Verbreitung. Floppy Disks werden kleiner und können gleichzeitig mehr Daten speichern. In zehn Jahren, seit 1981, reduzierte sich ihr Durchmesser von 5.25 auf 3.5 oder 2 Zoll, die Kapazität stieg von 360 Kilobytes auf über 2 Megabytes. War 1981 eine 5 Megabytes Festplatte für einen PC noch Luxus, so erscheinen ein Jahrzehnt später Normalkonfigurationen mit über 200 Megabytes, bei Workstations von mehreren Gigabytes. Beim RAM war die Steigerung noch größer: Im selben Zeitraum wuchs die typische Ausstattung von 64 Kilobytes auf 4 Megabytes, bei Workstations auf über 100 Megabytes.

Das Betriebssystem

Das Betriebssystem (Operating System) ist die Verbindung zwischen der Hardware und den Anwendungsprogrammen. Die Wahl des Betriebssystems entscheidet mit über die Nützlichkeit einer Applikation. Dies trifft besonders auf CAAD in der Ausbildung und in der Praxis zu, denn hier sind die Multi-Tasking (gleichzeitig zur Ausführung von verschiedenen Programmen fähigen) und Multi-User (gleichzeitig von verschiedenen Anwendern benutzbaren) Betriebssysteme klar im Vorteil gegenüber den Single-Tasking, Single-User-Systemen. Dies bedeutet nicht, daß in der täglichen Büroarbeit Computer mit den letztgenannten Betriebssystemen nicht effektiv eingesetzt werden können, doch gerade in der Architektur kann die Nutzung von Affinitäten von Betriebssystem und Entwurfstätigkeit interessant sein.

Das Betriebssystem arbeitet im Hintergrund und nimmt den Anwendern und Programmentwicklern viele mühselige Kommunikations- und Ordnungsaufgaben ab. Heute stehen für Architekturanwendungen vor allem drei Betriebssysteme im Vordergrund: DOS, UNIX und das Macintosh-Betriebssystem (Mac OS). UNIX war von Anfang an auf Multi-Tasking und Multi-User-Operationen ausgelegt, während DOS und das Mac OS als typische Single-Tasking, Single-User Systeme begannen. Inzwischen haben sich die Grenzen verwischt, und auch DOS und Mac OS bieten ein einfaches Multi-Tasking an. Ende der achtziger Jahre kamen das von IBM vertriebene OS/2 und Windows von Microsoft hinzu. DOS ist weltweit das zahlenmäßig dominierende Betriebssystem. Das Mac OS ist bei den graphischen

Anwendungen stark verbreitet. UNIX hat die größte Verbreitung im technisch-wissenschaftlichen Bereich.

Alle Betriebssysteme bieten inzwischen graphische Benutzeroberflächen an, die von verschiedenen Arten der Fenstertechnik Gebrauch machen und durch graphische Interaktion zum Beispiel das Kopieren von Dateien erlauben. Windows, Openlook und Motif sind die am weitesten verbreiteten graphischen Oberflächen der Betriebssysteme.

Vernetzung - Networks

Vernetzung ist die Verbindung verschiedener Computer zum Zweck der Kommunikation. Die Verknüpfung kann physischer Natur sein, zum Beispiel in Form von Kabeln, oder sie kann durch drahtlose Verbindung, zum Beispiel durch Laser oder Mikrowellen hergestellt werden. Bekannte Netzwerkkarten sind Ethernet, Tokenring, FDDI oder LocalTalk. Zusätzlich bedarf es spezieller Software, um die Kommunikation zu gewährleisten. Bekannte Netzwerkprotokolle sind IPX (Novell), TCP/IP (UNIX), und AppleTalk (Mac OS).

Die Anwendungsgebiete der Netzwerke sind vielfältig. Im einfachsten Fall können verschiedene Computer miteinander kommunizieren und Daten austauschen. Jede Flugreservierung, jede Abfrage von einem Geldautomaten nutzt diese Fähigkeit. Netzwerke können auch die Arbeit am gleichen Projekt an verschiedenen Orten durch verschiedene Personen ermöglichen. In diesem Fall müssen besondere Vorkehrungen für die Erhaltung der Integrität des Projektes getroffen werden, zum Beispiel durch

gestaffelte Zugriffsrechte. In einem weiteren Schritt ist ein Szenario denkbar, in dem mehrere Millionen Computer über ein Netzwerk zusammengeschlossen sind, welche mit entsprechender Software einen gewaltigen Rechner bilden, in dem ein in viele Unterprozesse aufgeteilter Prozeß parallel abläuft. Der Unterhalt von Netzwerken stellt hohe Anforderungen an das Personal. Verbindungen können unterbrochen werden, Programme können abstürzen, Viren ihr Unwesen treiben. Da die Maschinen meist weit voneinander entfernt sind, ist die Fehler-suche kompliziert. Für die Anwender ist besonders zu Beginn die Kommunikation über das Netzwerk eine oft beängstigende Erfahrung, denn die erste Kontaktaufnahme mit einer anderen Person über ein elektronisches Medium ist ungewohnt.

Computersprachen - Computer Languages

Computersprachen sind, wie natürliche Sprachen, Hilfsmittel zur möglichst effizienten und vollständigen Kommunikation. Sie bestehen aus Vokabeln und Regeln, nach denen Symbole zu komplexeren Bedeutungen zusammengesetzt werden können. Die heute verbreitete Computer-Architektur, auch unter der Bezeichnung 'von Neumann-Maschine' bekannt, zeichnet sich dadurch aus, daß sie Computersprachen mit sequentiellen Kontrollstrukturen unterstützt. Man kennt prozedurale (Pascal), funktionale (Lisp), modulare (Modula) [Wirth 1985], (Oberon) [Reiser 1992, Wirth 1988], objektorientierte (Smalltalk), Dataflow (Linda) und logische Sprachen (Prolog). Die von Neumann-Architektur unterstützt diese Sprachen mit unterschiedlicher Effizienz.

Durch schneller werdende Hardware entsteht wachsender Druck, mit weiterentwickelten Computersprachen auf neue Möglichkeiten zu antworten. Besonders die Entwicklung von Massiv-Parallel-Rechnern (Massive Parallel Computers) eröffnet die Chance, eine Computersprache auf hohem Niveau zu entwerfen, die problemorientiert ist und es erlaubt, technisch-wissenschaftliche Sachverhalte automatisch zu parallelisieren [Knudsen 1992]. Den Sinn und die Notwendigkeit dieser Bestrebung wird jeder erkennen, der je versucht hat, entwurfsorientierte Problemstellungen mit Hilfe von Computersprachen in Programme umzusetzen, die das Problem lösen. Ein Problem ist, daß die Programmentwicklung für Massiv-Parallel-Computer noch immer sehr komplex ist.

Programme - Software

Mit Software bezeichnet man die Programme, die zwischen Anwender und Maschine stehen. Auch das Betriebssystem und die Netzwerksoftware sind Programme. Man vergleicht Software oft mit menschlichen Gedanken, während man in der Hardware das Äquivalent zum menschlichen Hirn sieht. Beide Vergleiche sind anschaulich, doch zugleich irreführend. Der Entwurf und die Programmierung von Software hat sich zu einer Wissenschaft entwickelt, die viel Können verlangt. Architekturbezogene Software ist geprägt von der Eigenschaft, daß die Programmiererinnen und Programmierer in den seltensten Fällen Architekten sind. Dies führt unter anderem dazu, daß sich die wenigste CAD-Software zum Entwerfen eignet. Diese Tatsache hat allerdings auch den positiven Effekt, daß CAD-Software viele Mög-

lichkeiten bietet, an die man als traditionell gebildeter Architekt nie denkt.

Wie entsteht Software? Am Anfang stand die Idee, wiederkehrende numerische Probleme erst durch mechanische, dann durch elektronische Hilfsmittel zu lösen. Immer gab es dabei einen 'intelligenten' Teil, der eine weniger intelligente Maschine steuerte. Als frühe Beispiele sind Apparate zur Berechnung ballistischer Flugbahnen bekannt. Die Eingabe- und Ausgabeteile der frühen Geräte - die Peripherie - war daher zunächst zahlen- und buchstabenorientiert. Sobald Hardware und Software in der Lage waren, Bildschirme für die Darstellung von Graphik zu nutzen, begann die Beschreibung architekturorientierter Programme. Mit Recht gilt Evan Sutherlands Arbeit von 1963, allgemein unter dem Titel 'Sketchpad' (Skizzenblock) bekannt, als Ursprung der modernen CAD-Systeme [Sutherland 1963]. Man bedenke, es war die von einem Einzelnen formulierte Idee, die, obwohl nie vollkommen realisiert, gewaltige Wirkungen zeigte. Es dauerte ein Jahrzehnt, bis die ersten großen CAD-Programme entstanden, die bereits über fortschrittliche Datenstrukturen verfügten. Es dauerte ein weiteres Jahrzehnt, bis die ersten brauchbaren Mikrocomputer-CAD-Programme auftauchten.

Der Aufbau eines Massenmarktes hatte für die Softwareentwicklung zwei wichtige Konsequenzen. Zum einen mußten die Hersteller sich den Bedürfnissen der Anwender anpassen und implementierten so Programme, die so aussahen, als unterstützten sie den Entwurf. In Wahrheit gaben sie aber nur Hilfe bei der Eingabe des Ergebnisses des Entwurfs. Zum anderen wurde durch das wachsende Interesse an CAD schnell eine kritische Zahl von Anwendern

und Programmierern erreicht, die auch unkonventionelle Entwicklungen vorantreibt.

Anfang der neunziger Jahre zeigt sich mit wenigen Abweichungen in den Architekturbüros der Industrieländer ein differenziertes Bild. Textverarbeitungs- und Tabellenkalkulationsprogramme sind in fast allen Büros vorhanden. Zweidimensionale Zeichenpakete setzen sich langsam durch, weniger Büros besitzen und nutzen dreidimensionale Software. Die Architektinnen und Architekten, die Software zum Entwerfen erfolgreich einsetzen können, bilden noch eine sehr kleine Minderheit [Behaneck 1991]. Noch weniger Anwender beherrschen Software zur Durchführung von Simulationen oder multimedialen Präsentationen. Am wenigsten genutzt ist die anscheinend logischste Fähigkeit von Software, verschiedene Aspekte des Entwerfens und Bauens miteinander zu verknüpfen und in integrierten Programmen zu simulieren.

Benutzeroberflächen - User Interfaces

Benutzeroberflächen stehen zwischen Programm und Anwendern. Ihre Entwicklung seit Beginn der achtziger Jahre, zusammenfallend mit der Einführung graphischer Benutzeroberflächen, ist rapide. Dabei lassen sich zwei Hauptrichtungen unterscheiden:

(1) Anpassung an das Computermodell. Die Benutzeroberfläche wird so entwickelt, daß sie die Hauptfunktionen der Programme schnell, eindeutig und effizient ausführt. Beispiele sind CAD-Programme, in denen komplexe Handlungen durch kurze Befehle oder Funktionstasten ausgeführt werden können.

Dies ist zu Anfang zwar kryptisch, für geübte Anwenderinnen und Anwender aber sehr effizient.

(2) Anpassung an das Anwendermodell. Die Benutzeroberfläche wird so ausgelegt, daß sie die Gewohnheiten der Anwenderinnen und Anwender berücksichtigt und implementiert. Typisches Beispiel ist das 'Desktop Paradigm' oder Schreibtischparadigma. Dabei herrscht auf dem Bildschirm dieselbe Ordnung (oder Unordnung) wie auf dem Schreibtisch und durch Schieben und Klicken lassen sich die meisten täglichen Organisationsaufgaben durchführen.

Beide Richtungen haben Vor- und Nachteile. Die Anpassung an das Computermodell charakterisierte die frühen Programme. Die Anpassung an das Anwendermodell ist neueren Datums und für die Anwender von Vorteil. Das größte Problem ist hier natürlich die Definition und das Finden der typischen Anwenderinnen und Anwender, die es, wie sie sich schnell herausstellt, überhaupt nicht gibt. Das allgemeine Anwendermodell ist nur partiell vorhanden, und so ergeben sich für Benutzeroberflächen wie für technische Geräte immer wieder Probleme.

Das Entwerfen von Benutzeroberflächen hat sich bereits zu einem eigenen und umfangreichen Forschungsgebiet entwickelt und ist in verschiedene Richtungen gespalten. Forschungsinhalte sind unter den Abkürzungen UIS (User Interface Systems) und GUI (Graphical User Interface Systems), UIMS (User Interface Management Systems) sowie UIDE (User Interface Design Environments) bekannt [Carlsen 1991]. Die beste Benutzeroberfläche für den Entwurf, - aber auch

für die meisten anderen Programme - ist die, bei der der Computer 'verschwindet' und sich die Anwenderinnen und Anwender direkt auf das Problem konzentrieren können. Denn die meisten Menschen benutzen Computer, um Probleme besser lösen zu können. In seinem unterhaltsamen Buch 'The Psychology of Everyday Things' beschreibt Donald Norman, was bei der Interaktion mit einem Computer alles schief gehen kann:

"Haben Sie sich jemals vor einen typischen Computer gesetzt? Wenn ja, dann haben auch Sie die 'Tyrannei des leeren Bildschirms' erfahren. Die Person sitzt vor der Maschine, bereit zu beginnen. Was beginnen? Wie beginnen? Der Bildschirm ist entweder vollkommen leer oder gefüllt mit unverständlichen Worten oder Symbolen, die keinen Hinweis darauf geben, was als nächstes erwartet wird. Es gibt da eine schreibmaschinenähnliche Tastatur, aber keinen Grund zu der Annahme, daß eine Taste einer anderen vorzuziehen sei. Und überhaupt, ist es nicht so, daß ein falscher Tastendruck die Maschine zum Absturz bringen kann? Oder wertvolle Daten zerstört? Oder Sie zufällig mit einer hochgeheimen Datenbank verbindet und eine Untersuchung Ihrer Person durch den Geheimdienst zur Folge hat? Wer kennt die Gefahren, die hinter dem Drücken einer Taste lauern? Es ist fast so beängstigend, wie zu einer Party von Fremden geführt und dann in der Mitte ohne Begleitung stehengelassen zu werden. Ihr Gastgeber verschwindet und sagt: 'Fühlen Sie sich wie zuhause. Ich bin sicher, hier werden sie viele Leute treffen, mit denen sie sprechen können'. 'Ich nicht. Ich verziehe mich an den Rand und suche etwas zu lesen'." [Norman 1988, dt. v. Verf.]

Die Maschine im Architekturbüro: Computer Aided Architectural Design als Teil einer Strategie

Das neue Architekturbüro unterscheidet sich grundsätzlich von dem der Vergangenheit. Der Arbeitsplatz der Architektinnen und Architekten, einst die billigste aller vorstellbaren Arbeitsumgebungen in Einrichtung und Unterhalt, wird kapitalintensiv. Seit Mitte der achtziger Jahre tritt zudem vermehrt der Fall auf, daß die Mitarbeiterinnen und Mitarbeiter in einem Büro über mehr technisches Wissen und Gerät verfügen als die Chefs. Doch nicht nur im materiellen Bereich - Hardware und Peripherie - deuten sich Veränderungen an. Auch das intellektuelle Instrumentarium der Architektur in der Form neuer Programme und Modelliermöglichkeiten wächst ständig.

Grundlagen des CAD

Computer Aided Design (CAD) beruht auf einer neuen, bauspezifischen Arbeitsmethode, sowie auf einem System von Komponenten. Die Komponenten entwickelten sich aus Forschungsergebnissen, die in den letzten 30 Jahren auf verschiedenen Gebieten erzielt und anschließend in Software und Hardware umgesetzt wurden:

(1) Computersprachen. Die meisten CAD-Programme wurden in den zuvor beschriebenen Sprachen Pascal, Fortran und C entwickelt. Wachsender Beliebtheit erfreuen sich die objektorientierten Sprachen.

(2) CAD-Programme. Hier wurden die größten Fortschritte in den Benutzeroberflächen und in der Anpassungsfähigkeit gemacht. Verbesserungen gibt es in der Integration von CAD mit anderen für den Entwurf wichtigen Programmen. Die Grundfunktionalität hat sich dagegen wenig verändert.

(3) Modellieren. Dreidimensionales Modellieren bis hin zum Volumenmodellieren ist vom Mainframe über die Workstations in den PC-Bereich vorgedrungen, hat aber noch nicht die Eleganz und Geschwindigkeit zweidimensionalen Modellierens erreicht.

(4) Graphik. Die Algorithmen zur Darstellung von Objekten haben ein hohes Maß an Effizienz und Realismus erreicht. Graphikbibliotheken wie GKS, PHIGS, Open GL und HOOPS zur Erstellung und Manipulation zwei- und dreidimensionaler Flächenobjekte sind auf verschiedensten Hardware Plattformen verfügbar.

(5) Rendering. Verschiedene Schattierungsalgorithmen, wie Phong, Gouraud, und Texture-Mapping für die photorealistische Darstellung, sind bereits als Hardware implementiert, und schnelle Strahlen-Verfolgungs- (Ray Tracing)- und Radiosity-Verfahren, die unter anderem scharfe und weiche Schatten darstellen können, werden in Form von Softwarebibliotheken allgemein zugänglich.

(6) Datenbanken. Relationale Datenbanken haben sich kommerziell durchgesetzt. CAD-Programme benutzen aus Effizienzgründen für ihre Graphik allerdings meist nicht-standardisierte oder proprietäre Datenstrukturen. Die Verknüpfung mit leistungsstarken, kommerziellen Datenbanken, ein für die Entwicklung integrierter Programme unabdingbarer Schritt, vollzieht sich allmählich auch bei kleineren Programmen.

(7) Prozessoren. Neue CPUs verarbeiten Information mit immer größerer Geschwindigkeit. RISC (Reduced Instruction Set Computing) -Prozessoren dominieren bei Workstations, auf PCs sind die CISC (Complex Instruction Set Computing) -Prozessoren vorherrschend.

Von der Funktionalität des Gesamtsystems hängt die Einsatzbarkeit im Architekturbüro ab. Der Fortschritt in der Software ist langsamer als die Weiterentwicklung in der Hardware, doch sind die Risiken, die man heute mit der Einführung von CAD im Büro eingeht, bei weitem nicht mehr so groß wie noch vor zehn oder 20 Jahren. Es ist zum Glück so, daß heute ein CAD-System mehr Konzentration auf die Architektur erlauben als in der Vergangenheit.

Neue Chancen für das Architekturbüro

Chancen bietet CAD auf den Ebenen Organisation, Produktion, Integration und Kreativität. Alle diese Möglichkeiten sind zugleich Gefahren, wenn sie nicht richtig genutzt werden:

(1) Organisation. Der erste wahrnehmbare Effekt bei der Einführung von CAD ist eine Änderung der Bürostruktur und der Organisation von Daten und Zeichnungen. Wenn die Architektinnen und Architekten selbst das neue Mittel beherrschen und verwenden, sind die Änderungen am geringsten. Falls sie nicht mit dem neuen Medium vertraut sind, werden sie von der direkten Interaktion mit den Plänen durch die neu eingeführte Ebene Computeroperator oder CAD-Zeichner getrennt, eine langfristig verhängnisvolle Entwicklung. Auch die Organisation der Dokumente ändert sich: Zu Beginn werden Pläne und Computermodelle vielfach dupliziert und existieren parallel. Die Einführung von CAD erfordert also eine neue Disziplin, nicht unbedingt mehr Disziplin bei der Organisation als bisher. Die Vorteile der neuen Organisation werden sich schnell zeigen: weniger Fehler, weniger Duplikation, mehr Überblick und bessere Kontrolle des Ergebnisses.

(2) Produktion. Heutige CAD-Programme sind für die Produktion von Dokumentationen optimiert. Im zweidimensionalen Bereich haben einzelne Programme inzwischen eine Benutzerfreundlichkeit und Effizienz erreicht, die das noch vor wenigen Jahren gültige Argument, "man könne das mit der Hand schneller", entkräften. Gut ausgebildete, mit einem System vertraute Zeichnerinnen und Zeichner lassen den manuellen Methoden bei der Produktion von Zeichnungen keine Chance. Bei Präsentationszeichnungen und im dreidimensionalen Bereich bestehen allerdings noch Schwächen der CAD-Programme.

(3) Integration. Dies ist eine der potentiell größten, aber bisher am wenigsten genutzten Vorteile der neuen Technologie. Dadurch, daß auf der untersten Ebene alle Informationen in einem digitalen Modell repräsentiert sind, ist eine Kompatibilität auch auf höherer Ebene möglich. So lassen sich Text, Graphik und Zahlen in eindeutiger Weise miteinander in Beziehung setzen. Die entsprechenden Verknüpfungen sind ständig abrufbar und mit dem Computermodell gespeichert. Sie müssen nicht jedesmal neu mühsam hergestellt werden, wie dies bei der Verwendung traditioneller Methoden der Fall ist.

(4) Kreativität. Mit der Erfahrung wächst die Chance, daß die Arbeit mit dem neuen Instrument zu mehr und verantwortungsvollerer Kreativität führt. Besonders unterstützt wird Kreativität bei der sorgfältigen Evaluation und Verbesserung bestehender Lösungen, aber auch beim Testen völlig neuer Ideen. Durch Simulation und Evaluation lassen sich unangenehme Überraschungen vermeiden. Die Bauherrschaft hat die Chance, daß ein in der Simulation durch verschiedene Tests gelaufenens Gebäude weniger Probleme aufweist als ein traditionell entworfenes Bauwerk.

Gesichtspunkte bei der Anschaffung eines CAD-Systems

Berufsverbände und private Beratungsfirmen in allen industrialisierten Ländern veröffentlichen Checklisten, die bei der Anschaffung einer CAD-Anlage zu beachten sind. Diese ändern und vervollständigen sich jedes Jahr und sollen deshalb hier nicht dupliziert werden. Unabhängig davon muß man sich zu Beginn über die folgenden Faktoren ein klares Bild verschaffen [Meißner 1992]:

(1) Die grundsätzliche Einstellung der Büroleitung zum Thema CAD. Die Leitung muß ganz hinter dem Projekt stehen, die entsprechenden Ressourcen, von denen Hard- und Software nur ein kleiner Teil sind, besitzen und für diesen Zweck freigeben.

(2) Der Wissensstand der Büroleitung zum Thema CAD. Je höher der Wissensstand auf Seiten der Entscheidungsträger, um so entspannter sind die Diskussionen mit den Anbietern und um so mehr kann man sich auf das anstehende Problem konzentrieren.

(3) Der Wissensstand der Mitarbeiterinnen und Mitarbeiter. Im CAD-Bereich haben junge Mitarbeiter oft einem höheren Wissensstand als die, von denen sie beschäftigt werden. Dies ist vollkommen akzeptabel, doch darf sich daraus keine Abhängigkeit entwickeln, was bedeutet, daß das Wissensgefälle nicht zu stark sein sollte.

(4) Das zur Lösung anstehende Problem. Handelt es sich um ein Projekt, als dessen Teil die CAD-Umgebung angeschafft werden kann oder muß, läßt sich Hilfe von den Projektpartnern erhalten. Handelt es sich um den ohnehin geplanten Einstieg in CAD, ist eine eigene Evaluation notwendig.

Bei der Anschaffung der ersten CAD-Anlage ist ein pragmatischer, nicht dogmatischer Ansatz ratsam. Das Vorhandensein einer Strategie ist wichtig, in die

danach die heute existierenden Insellösungen als Teil eines Ganzen integriert werden können. Neben der Beratung durch Hardware- und Software-Vertreter ist immer ein Erfahrungsaustausch mit unabhängigen Beratern und bereits CAD-erfahrenen Berufskolleginnen und Kollegen zu empfehlen.

Unabhängig von den speziellen Gesichtspunkten ist auf jeden Fall auf Kompatibilität zu achten, besonders wenn zu Anfang verschiedene, nicht integrierte Programme verwendet werden. Gleiches Betriebssystem und gemeinsame Datenaustauschformate sind die minimalen Anforderungen. Als Alternative zum gleichen Betriebssystem sind Programme akzeptabel, die auf verschiedenen Hardware-Plattformen das gleiche und austauschbare Datenformat erzeugen. Sind mehr als zwei Maschinen geplant, ist unbedingt die Vernetzung vorzusehen.

Die Schulung der Mitarbeiterinnen und Mitarbeiter

Die Schulung ist noch immer der gewichtigste Grund für Erfolg oder Mißerfolg der Einführung von CAD. Die Technologie ist noch nicht auf dem Stand, daß sie ohne Vorbereitung zur Verwendung kommen kann. Schulung ist auf den Ebenen Integration in die Büroumgebung, Software und Hardware notwendig. Der letzte Punkt verliert dabei mit der Zeit an Bedeutung, während die beiden ersten immer wichtiger werden. Noch vor wenigen Jahren war die Reihenfolge fast umgekehrt. Drei grundsätzliche Möglichkeiten bieten sich an:

(1) Die Mitarbeiter bringen bereits Erfahrung mit, die sie sich an der Hochschule, der Fachhochschule, in privaten Schulungseinrichtungen oder im Büro angeeignet haben. Damit liegt der Schwerpunkt auf der

sinnvollen Integration der Kenntnisse der neuen Mitarbeiter in der eigenen Organisation.

(2) Die Schulung von Mitarbeiterinnen und Mitarbeitern durch Fort- und Weiterbildung. Damit ist eine allmähliche Einführung von CAD in die eigene Struktur möglich und erlaubt einen relativ weichen Übergang. Nachteil ist die fehlende Erfahrung zu Beginn der Einführung von Computern im Büro.

(3) Die interne Schulung. Besonders bei großen Firmen, die gegebenenfalls sogar eigene Softwareentwicklung betreiben, ist diese Art der Schulung verbreitet. Sie bindet die Mitarbeiter einerseits mehr an die Firma; andererseits ist die Flexibilität geringer und der Overhead für die Ausbildung kann beachtlich sein. Beispiele sind die Firmen SOM (Skidmore, Owings and Merrill) in den USA oder Suter+Suter in der Schweiz. Einige Firmen mußten deshalb in letzter Zeit auf diese Möglichkeit verzichten und auf die beiden ersten zurückgreifen.

Bis in die jüngste Vergangenheit existierte das Problem, daß die intern geschulten Mitarbeiterinnen und Mitarbeiter eine spezielle Stellung in der Firma genossen, da sie zum Teil beträchtliche Investitionen erforderten. Inzwischen ist die erste Art der Schulung zahlenmäßig dominierend, so daß CAD-Erfahrung nur noch bedingt als Verhandlungsvorteil bei der Einstellung einsetzbar ist. Mehr noch, CAD-Erfahrung wird immer mehr zur unabdingbaren Voraussetzung für eine Anstellung in vielen Architekturbüros.

Ökonomische Aspekte - Kosten und Nutzen

Diese Frage interessiert alle, die CAD aus Rationalisierungsgründen einführen. Exakt läßt sich diese Frage, ob und wann ein Büro durch die Einführung von

CAD in die Gewinnzone kommt, nicht beantworten, doch kann man von folgenden Erfahrungswerten ausgehen: In den ersten drei Monaten nach der Einführung ist mit einem Produktivitätsrückgang zu rechnen, nach fünf bis sechs Monaten ist die ursprüngliche Produktivität wieder erreicht. Danach steigt die Leistung deutlich an und stabilisiert sich je nach Arbeitsbereich und Fähigkeit auf einem Niveau von 150-200 Prozent. Diese Betrachtungsweise berücksichtigt nicht die Veränderungen des Produktionsprozesses selbst, den die Verwendung des Computers bewirkt.

Die kurzfristigen Produktionssteigerungen um so höher, je besser das CAD-Programm die bisherigen manuellen Tätigkeiten ersetzt. Langfristig sind höhere Steigerungen aber nur möglich, wenn sich der Integrationsgrad zwischen den Teilarbeitsbereichen erhöht. Das bedeutet, daß die Ersetzung des bestehenden Zeichengeräts durch das CAD-Programm in Form eines elektronischen Zeichenstifts, obwohl zuerst verlockend, relativ schnell an ökonomische Grenzen stößt. Es bedeutet weiterhin, daß nur eine langfristige Strategie, bestehend aus Schulung, Organisation und Computereinsatz, das ökonomische Potential voll ausschöpfen kann.

Als Fazit gilt, zuerst die kostenintensivsten Bereiche zu identifizieren und danach zu klären, welche Software diesen Bereich am besten unterstützt. Erst dann soll die Wahl der Hardware erfolgen. Hans Kahlen bemerkt dazu ganz richtig und lapidar: "Theoretische Produktivitätsgewinne durch den Einsatz von EDV auf einem bestimmten Feld werden sich nämlich nur dann in einer signifikanten Zeitersparnis auswirken, wenn dieses Feld einen wesentlichen Teil des Arbeitsumfangs eines Büros ausmacht; Zeitersparnis läßt

sich aber nur dann in eine höhere Einnahme umsetzen, wenn entsprechende zusätzliche Aufträge akquiriert werden können. Um eine EDV-Anlage zu finanzieren, müssen jedoch, wenn sich das nicht in einer Erfolgsminderung niederschlagen soll, entweder

mehr bzw. höhere Honorare erzielt, oder Aufwendungen vermindert werden. Es sei denn, der Büroinhaber sei bereit, dafür zu bezahlen, daß mehr Zeit zur Verfügung steht, um die vorhandenen Aufträge abzuwickeln" [Kahlen 1989].

Die neuen Instrumente der Architektur: CAAD-Komponenten und -Programme

Bei der Betrachtung der Grundlagen von CAD und bei der Vorstellung der Komponenten und Programme sind zwei Perspektiven zu unterscheiden. Die erste geht vom Ist-Zustand aus, der auf dem CAD Gebiet von einer großen Zahl kommerzieller Insellösungen für spezielle Probleme gekennzeichnet ist. Alle, die schnell eine CAD-Umgebung brauchen, müssen sich mit diesem Ist-Zustand auseinandersetzen und die entsprechenden Komponenten auswählen. Die zweite Betrachtungsweise ist zukunftsorientierter und geht davon aus, daß mittelfristig die verschiedenen Komponenten besser kompatibel werden und langfristig in einem integrierten System zur Verfügung stehen werden. Mit dieser Möglichkeit vor Augen, wird die Planung der Infrastruktur, die Anschaffung der Maschinen und Software sowie die Ausbildung der Beteiligten anders aussehen.

Hardware für die Architektur

Für das durchschnittliche Architekturbüro kommen heute an Hardware hauptsächlich die Alternativen Workstation oder PC in Frage, wobei beide sich in Leistungsfähigkeit und Preis annähern. Die größten Leistungengpässe gibt es und wird es weiterhin im Graphik- und besonders im 3D Bereich geben, sowie in der schnellen Verwaltung und Verknüpfung der riesigen Datenmengen in einem integrierten System. Obwohl die Hardware beim Anschaffungspreis der CAD-Umgebung noch eine wichtige Rolle spielt, muß sie als Teil eines Ganzen gesehen werden. Die Hardware steht in direktem Verhältnis zum Inhalt der Aufgabe. Textverarbeitung und dreidimensionale Animation verlangen so verschiedene Leistungsmerkmale, daß man sie kaum auf derselben Hardware Plattform

ausführen wird. Besondere Sorgfalt ist bei der Graphik geboten, denn Graphiksoftware hat oft sehr spezielle Hardwareanforderungen. Oft läßt sich die entsprechende Graphikleistung durch eine Graphikkarte verbessern, die auf zwei- und dreidimensionale Operationen optimiert ist. Für die Systemintegration dieser speziellen Graphik-Komponenten ist jedoch entsprechendes Know-How notwendig.

Programme für die Konstruktion

Konstruieren geschieht im zwei- und im dreidimensionalen Raum. Ganz bewußt wollen wir uns hier auf den Teil der Konstruktion beschränken, der auf ein bestehendes Konzept zurückgreift und lediglich die räumliche Ausformung des Konzeptes beinhaltet. Konstruktion ist ein natürliches Einsatzgebiet für Computer, denn die Theorie ist eindeutig formuliert und in Form von Programmen implementiert, die Aufgabenstellungen sind klar. Das neue Werkzeug hebt die Grenzen zwischen Geometrie und Algebra auf und erlaubt schnelles und exaktes Konstruieren. Es bleibt die Tatsache, daß die wenigsten CAD-Applikationen unter Einbezug von Entwurfs-Architekten geschrieben sind. Die Autoren der Programme sahen das Endprodukt eines architektonischen Prozesses und haben dieses nachvollziehbar gemacht. Der Prozeß selbst aber findet dadurch kaum Unterstützung; die Anwender müssen sich an das Programm anpassen. Erst in letzter Zeit setzen sich architekturenspezifische Programme durch, an deren Konzeption Architekten maßgeblich beteiligt waren.

Ein großer Vorteil der CAD-Programme ist, daß sie die Grenzen zwischen Skizzen, Projekt- und Ausführungsplanung aufheben können und theoretisch

die Arbeit an einem sich kontinuierlich weiterentwickelnden Modell denkbar wird. Oft sind die Teilnehmer am Entwurfs- und Konstruktionsprozeß aus geographischen oder organisatorischen oder fachtechnischen Gründen voneinander getrennt. Dadurch wird diese Integrationsmöglichkeit in den seltensten Fällen genutzt, und die Konstruktion wiederholt viele unnötige Schritte. Doch selbst mit diesen Handicaps ist die mit CAD mögliche Konstruktionserleichterung mit anderen Mitteln unerreichbar.

Datenbanken

Datenbanken sind die Bibliotheken und Archive des CAD. Sie sind zugleich flexibler und beschränkter als herkömmliche Speichermedien - flexibler, da sie aktive Beziehungen zwischen den einzelnen Daten herstellen und so neue Information erzeugen können, beschränkter, da sie bisher an das interaktive Ausgabemedium Bildschirm gebunden sind, der eine wesentlich geringere Auflösung und vor allem eine erheblich geringere Größe als herkömmliche Zeichnungen aufweist. Datenbankmanagement Systeme (Database Management Systems, DBMS) organisieren die Datenbanken. Mit strukturierten Abfragesprachen (Structured Query Languages, SQL) lassen sich die in Tabellen abgelegten Daten (Ordnungsprinzip der relationalen Datenbanken) nach Informationen durchsuchen. In kommerziellen Anwendungen in Banken und Versicherungen kommen relationale Datenbanken häufig zum Einsatz. Datenbanken sind in jedem CAD-System wichtig, denn alle CAD-Systeme benötigen für die Speicherung der Objekte einschließlich ihrer Attribute eine interne Struktur. Normalerweise ist diese Datenbank nur vom CAD-Paket aus zugäng-

lich, doch kann man sie bei einigen Programmen auch in externe, kommerzielle Datenbanksysteme exportieren.

An der Frage, wie geometrische Strukturen und zugehörige Text- und Bildinformation zu speichern und zu verwalten sind, scheiden sich die Geister. Relationale Datenbanken sind gut geeignet für die Speicherung allgemeiner Daten, für die Verwaltung graphischer Datenstrukturen sind sie dagegen weniger vorteilhaft. CAD-Programme haben normalerweise die entgegengesetzten Eigenschaften: Sie verwalten geometrische Informationen sehr effizient, sind aber schwach auf der nicht-geometrischen Seite. Ein integriertes und daher zukunftsorientiertes CAD-System muß jedoch auf beide Arten von Informationen schnell zugreifen können. Viel Hoffnung setzt man zur Zeit in objektorientierte Datenbanken, die ein intuitiveres Speichern und Abfragen von baurelevanten Aspekten erlauben werden.

Bereits jetzt stellen die meisten CAD-Programme immer wiederbenutzte, zum Teil bereits parametrisierte Elemente in internen oder externen Datenbanken zur Verfügung. Möbel, Treppen, oder jede andere Art standardisierter Elemente sind so erhältlich und können bei Bedarf in das CAD Modell eingesetzt werden. Die mit diesen Elementen verbundenen Attribute sind auch in internen oder externen Datenbanksystemen weiterverarbeitbar. So lassen sich zum Beispiel alle Telephonanschlüsse mit den zugehörigen Telephonnummern der Mitarbeiterinnen und Mitarbeiter in einem Gebäude anzeigen, oder die Kosten aller Einrichtungen auflisten.

In diesen großen Möglichkeiten liegt aber auch zugleich die Gefahr der Datenbank. Sie ist nur vollständig, wenn jedes im CAD-Modell erscheinende Element auch wirklich Attribute besitzt, die in der

Datenbank erfaßt sind. Und sie ist nur sinnvoll, wenn jede Änderung in der Planung oder am bestehenden Gebäude in der Datenbank vermerkt wird, denn sonst werden Entscheidungen auf falschen Voraussetzungen getroffen. Damit sind bereits die Möglichkeiten und Probleme des Facility Management angesprochen, denn auch dort ist eine vollständige und aktuelle Datenbank Voraussetzung für sinnvolle Arbeit mit der neuen Technik.

Mit Blick auf die Zukunft sind die Bestrebungen zu einer Integration aller beim Planen und Bauen benötigten Informationen sehr wichtig. Immer mehr erweist sich - die übrigens schon aus den Anfängen des CAD stammende - Erkenntnis als richtig, daß nur über eine Datenbank die Beziehungen zwischen Geometrie und Erstellungskosten oder zwischen Energieverbrauch und Erstellungskosten und statischem System sinnvoll zu verknüpfen sind. Damit wird das CAD-Programm eines in einem Set von Programmen, die an einem Projekt zum Einsatz kommen, verknüpft über eine Datenbank.

Integrierte Programme

Traditionell war die Textverarbeitung in allen Berufen der erste Bereich, mit dem der Computer in das Büro einzug hielt. Inzwischen steht in fast jedem Büro eine Maschine für solche Zwecke, was aber nicht bedeutet, daß sie auch optimal eingesetzt wird. Da die bisherige Bürokommunikation auf dem Medium Papier beruhte, haben Computer und Programme dieses Medium zuerst imitiert und die Herstellung von Papierdokumenten vereinfacht. Dies erklärt die an sich erstaunliche Tatsache, daß nach der Einführung von Computern, die das 'papierlose Büro' ermögli-

chen sollten, der Papierverbrauch um bis zum Doppelten gestiegen ist. Denn Computer mit Textverarbeitungsprogrammen werden meist noch wie sehr schnelle Schreibmaschinen verwendet.

Gute Textverarbeitungsprogramme bieten Hilfe für die Rechtschreibung, die Trennung, die Graphik, den Seitenumbruch, die Gestaltung, sowie die Möglichkeit, den Text in Abschnitte zu gliedern und in der Gliederung Hierarchien einzuführen. Diese eignen sich hervorragend für das Erstellen von langen Dokumenten und Beschreibungen, die allerdings im Baubereich seltener gebräuchlich sind. Für die immer wieder vorkommenden Ausdrücke und ganze Paragraphen gibt es Textbausteine, also Textabschnitte, die über eine Kurzbeschreibung abgerufen und dann in voller Länge in das Dokument eingesetzt werden.

Tabellenkalkulationsprogramme, eine wirkliche Neuerung und erst durch den Computer sinnvoll und möglich geworden, sind eine ausgezeichnete Rechenhilfe, mit der sich komplexe numerischen Zusammenhänge schneller als in der Vergangenheit darstellen und lösen lassen. Verbunden mit graphischen Darstellungen in Form von Balken- und ähnlichen Diagrammen sind sie zum ausgezeichneten Analysewerkzeug geworden. Durch die Möglichkeit, auch 'What-If' Rechnungen durchzuführen, also Hypothesen sehr schnell zu testen, gewinnen sie sogar eine gewisse Bedeutung für den Entwurf.

Die notwendige Integration zwischen den Applikationen findet erst langsam statt, aber es gibt bereits eine Reihe ausgereifter integrierter Programme. Diese verbinden zum Beispiel ein Text- und ein Tabellenkalkulationsprogramm, so daß sich jede Änderung in den Zahlen direkt im Text zeigt. Ein weiteres Beispiel ist die Integration von Datenbank, Tabellenkalkulation und Text, so daß man in dem Bereich arbeitet, der

momentan am meisten interessiert, aber trotzdem ständig auf die anderen Bereiche zurückgreifen kann. Als drittes Beispiel mag die Integration von CAD-Programm, Datenbank und Spreadsheet dienen, die zumindest im Ansatz sogar in kleinen Programmen vorhanden ist. Dabei zeigen sich die numerischen Resultate der Änderungen im Entwurf sofort in der Datenbank und im Spreadsheet.

Für Büros, die sich mit der Erstellung von Gutachten beschäftigen oder allgemein viel mit Integration von Graphik und Text zu tun haben, bieten sich als Alternative zum traditionellen Einkleben der Bilder verschiedene Desktop Publishing (DTP)-Programme an. Solange die Datenflüsse eindeutig definiert sind, liefern die integrierten Programme gute Ergebnisse und helfen, Zusammenhänge besser zu verstehen oder gar erst zu erkennen. Sie versagen, wenn man versucht, die Analyse in eine Synthese zu verwandeln, also zum Beispiel den simulierten Energiebedarf eines Gebäudes im Tabellenkalkulationsprogramm zu ändern und zu erwarten, daß das CAD-Programm die entsprechenden Änderungen am Gebäude vornimmt.

Das elektronische Skizzenbuch

Warum sollte man ein elektronisches Skizzenbuch benutzen, wenn gerade diese Tätigkeit am besten mit der Hand geschieht? Und gibt es so etwas überhaupt bereits? Die Hersteller von Notebooks, Penbooks und Kleincomputern werden diese Frage bejahen, in der Praxis aber werden sich beim gegenwärtigen Stand der Technik wenige Architektinnen und Architekten mit dieser Idee anfreunden können.

Das elektronische Skizzenbuch ergibt unter drei Voraussetzungen Sinn. Erstens muß es so leicht und

transportabel sein, daß es überallhin mitgenommen werden kann und jederzeit zum Aufschreiben oder Aufzeichnen von Ideen zur Verfügung steht. Dies ist eine Frage der Technologie und der Benutzeroberflächen. Zweitens muß die Software so intelligent sein, daß sie die Skizzen objektorientiert speichert, mit Attributen versehen und in einer Datenbank ablegen kann. Drittens sollte die Software in der Lage sein, die Skizzen in entsprechende Modelle zu überführen und dort weiterzuverarbeiten.

Sind diese drei Bedingungen nicht erfüllt, ergibt sich kaum ein Vorteil gegenüber der Verwendung des Mediums Papier, zumal man bereits jetzt Skizzen einfach einscannen und danach weiterverarbeiten kann. Auch muß man das Skizzieren auf dem Computer erst erlernen. Die dazu zur Verfügung stehende Hard- und Software mit drucksensitiven Tablettts und drahtlosen Stiften ist vielversprechend. Obwohl Architekten mit der Qualität der heute angebotenen Pen-Technologie kaum zufrieden sein können, werden die zuvor genannten drei Bedingungen in wenigen Jahren erfüllt sein.

Rendering und Animation

Die Animation ist einem vierdimensionalen Modell vergleichbar, mit den drei herkömmlichen Dimensionen Punkt (0), Linie (1), Fläche (2) und Volumen (3) sowie der zusätzlichen Dimension Zeit. Eine ausführliche Beschreibung der verschiedenen Techniken findet sich bei Thalmann [Thalmann 1991]. Auf den amerikanischen SIGGRAPH Konferenzen und auf der europäischen Ars Electronica werden die jeweils letzten Entwicklungen vorgestellt.

Animation ist neben der virtuellen Realität heute die aufwendigste, aber auch eine effektive Form der Präsentation. Architektur-Animationen scheinen ein Widerspruch in sich, ist doch die Architektur meist statischer Natur. Deshalb zeigen die meisten Architektur-Videos nicht das Gebäude als bewegt, vielmehr bewegt sich der Betrachter um und durch das Objekt. Um den Eindruck einer weichen, nicht ruckartigen Animation zu erzeugen, benötigt man 25 bis 30 Einzelbilder oder Frames pro Sekunde, für deren Herstellung es zwei grundsätzliche Möglichkeiten gibt:

(1) Real-Time - Animation. Hauptvorteil ist die vollkommene Kontrolle des Betrachters, der sich ungestört im und um das Gebäude bewegen kann. Alle Bewegungssequenzen lassen sich in Echtzeit auf Video aufzeichnen und sind danach editierbar. Ein Nachteil ist, daß damit möglicherweise unbeabsichtigte Dinge entdeckt werden, die bei der traditionellen Art der Animation erst gar nicht erzeugt oder weggeschnitten würden. Ein weiterer Nachteil ist die erforderliche hohe Rechnerleistung und die damit verbundenen Kosten.

(2) Traditionelle Animation. Dabei werden die einzelnen Frames gerechnet, abgespeichert und anschließend auf Video aufgezeichnet. Die Geschwindigkeit des Prozessors ist dabei weniger kritisch als bei der Real-Time-Animation, da zum Rechnen auch die Nächte und Wochenenden genutzt werden können. Stehen sehr schnelle Festplatten oder große Mengen Hauptspeicher zur Verfügung, ist auch eine direkte Abspielung auf dem Computer denkbar. Nachteile sind das wesentlich kleinere Maß an Kontrolle: Ist erst der Animationsweg einmal festgelegt, rechnet der Computer stundenlang. Bei etwaigen späteren Änderungswünschen ist diese Rechenzeit fast immer verlo-

ren. Ebenso ist die Interaktivität der Real-Time-Animation mit dieser Technik unerreichbar.

Eine zunehmende Zahl von CAD-Programmen bietet einen Animationsmodul an, in dem das während des Entwurfs entstandene dreidimensionale Modell direkt kontrolliert werden kann. Dies ist die eleganteste und zukunftsichere Art der Animationsherstellung, denn dadurch werden Informationen nicht doppelt oder dreifach erzeugt.

Die Gräben zwischen zweidimensionaler Zeichnung, dreidimensionalem Modell und Animation werden kleiner, und es ist absehbar, daß bald zwischen dreidimensionalem Modell und Animation ein leichter Übergang möglich sein wird.

Ein CAAD-System

Ein CAAD-System besteht aus aufeinander abgestimmten Komponenten. Über die Jahre haben sich die Systeme um neue Peripherie erweitert, doch stärkere Veränderungen sind in den Komponenten selbst zu beobachten. Architektinnen und Architekten wird es am meisten interessieren, wie ihr eigenes System aussehen soll, doch darauf gibt es keine einfache Antwort. Jede seriöse Beratung wird zu dem selben Schluß kommen und den Büros individuelle Lösungen empfehlen. Doch einige Verallgemeinerungen sind zulässig.

(1) Das individuelle CAAD System. Gedacht für Einsteiger oder Architektinnen und Architekten, die auf Vernetzung keinen Wert legen. Es besteht aus einem PC mit erweiterbarer Speicherkapazität, einem mindestens 16 Zoll großen Farbmonitor, einem Flachbett Scanner, einem Laserprinter und einem Plotter.

(2) Das portable CAAD-System. Es besteht aus einem schnellen Notebook, möglicherweise mit eingebautem Modem, und einem portablen Printer. Die Einsatzmöglichkeiten sind begrenzt, aber die Kompatibilität mit einem größeren System über Datenaustauschformate ist gegeben, so daß sich das portable System auch für Detaillösungen eignet.

(3) Das vernetzte CAD-System. Bekannt als Client-Server-System, enthält es alle Komponenten des individuellen CAD Systems, doch wird das Betriebssystem eher UNIX sein. Es kann teure und leistungsfähige Peripherie mit anderen Computern im Netz teilen. Am wichtigsten aber ist die Fähigkeit, daß alle Anwenderinnen und Anwender im vernetzten System auf alle relevanten Datenbanken zugreifen und schnell untereinander Daten und Information austauschen können.

(4) Das vernetzte Rendering- und Animationssystem. Es erzeugt oder übernimmt über das Netz dreidimensionale Modelle, die danach als photorealistische Einzelbilder oder als Bildfolgen für Animationen gerechnet werden. Es hat Dia-Belichter und Videorecorder als zusätzliche Peripherie.

(5) Das vernetzte Multimedia-System. Es enthält alle Komponenten des vernetzten CAD- Systems und wird für die Produktion von Multimedia-Präsentationen benötigt. Spezielle Rendering und Animationssoftware, Ein- und Ausgänge für Audio und Video, eine Videokamera sowie ein digitaler Schnittplatz

ergänzen das System. Die Speicherkapazität für dieses System muß sehr groß sein.

(6) Das Virtual Reality-System. Aufbauend auf den Komponenten des Multimedia-Systems enthält es zusätzliche Hardware für die Berechnung zweier phasenversetzter perspektivischer Renderings, die den Blickwinkeln der beiden Augen entsprechen. Eine synchronisierte LCD-Shutter-Brille zeigt diese Bilder abwechselnd, so daß das Gehirn in der Lage ist, diese zu einem homogenen stereoskopischen Abbild zusammenzusetzen. In einer weiteren Ausbaustufe wird mittels Virtual Reality-Schnittstellen ein noch tieferes Eintauchen in das Datenmodell ermöglicht. Dazu gehört auch die Einbeziehung von räumlich definierten Audioquellen. Damit wird es möglich sein, intuitive Designentscheidungen vorzunehmen, die auch auf der Darstellung von bisher im Planungsprozeß nicht direkt erfahrbaren Parametern beruhen können.

Alle Zwischenstufen und Kombinationen sind je nach Bedarf möglich. Die Kompatibilität zwischen den Systemen muß gewährleistet sein, um die einmal gemachten Investitionen in die Komponenten, vor allem aber die Zeit, die in die in den Komponenten gespeicherten Projekte investiert wurde, zu sichern. Über die einzelnen Komponenten, besonders aber zur CAD- Software, gibt es jährliche Zusammenfassungen [Henze 1992].

Maschine und Künstliche Intelligenz (KI)

Heute gehören Kameras, deren Chips mit unscharfer Logik (Fuzzy Logic) und neuronalen Netzen (Neural Network) arbeiten, zu den normalen Konsumgütern. Autofokus und garantiert scharfe Bilder sind das Resultat. Die Grundlagenforschung, die heute Produkte dieser Art ermöglicht, geschah um die Mitte des Jahrhunderts. Bereits Jahre, bevor Evan Sutherland seine Gedanken zu Sketchpad niederschrieb, kam der Begriff der KI 1956 nach einer Konferenz in Dartmouth in Umlauf. Damals wollte eine Gruppe junger Forscher in einer zweimonatigen Studie nachweisen, daß jede Art des Lernens und jeder andere Aspekt der Intelligenz so präzise beschrieben werden können, daß eine Maschine konstruierbar ist, die dieselben Vorgänge simuliert. 30 Jahre später definieren Eugene Charniak und Drew McDermott den Begriff KI folgendermaßen: "Artificial intelligence is the study of mental faculties through the use of computational models." [McDermott 1987] - "Künstliche Intelligenz ist die Erforschung mentaler Fähigkeiten mit Hilfe von Computermodellen." (dt. v. Verf.). Die an sich wenig kontroverse Definition basiert allerdings auf der fundamentalen Voraussetzung, daß das menschliche Gehirn auf zumindest einer Ebene wie ein Computer funktioniert. Diese implizite Annahme ist es wohl, welche die meiste Kritik und Verurteilung von KI nach sich zieht, ebenso wie die ursprünglich hochgesteckte Absicht, alle mentalen Prozesse wie Lernen und Intelligenz vollkommen zu erforschen und offenzulegen. Die Grundannahme scheint auch eine nahe Verbindung von KI und Psychologie zu indizieren. Jedoch sehen die meisten KI-Forscher ihr Feld als eigenständig an, müssen sie doch stets funktionierende und nachprüfbar Programme liefern, während in der Psychologie die experimentelle Bestätigung oft ausreicht.

Entwickler von KI-Programmen versuchen Probleme zu lösen, an denen herkömmliche Programmieransätze scheitern. Sie arbeiten mit Wissen, repräsentiert in verschiedenster Form, von einfachen heuristischen Regeln bis hin zu komplexen Objekten und Prototypen. Verallgemeinernd läßt sich sagen, daß die Erfolge der KI um so überzeugender ausfallen, je präziser Anwendungsgebiet und Problemstellung eingegrenzt sind. Die bekanntesten Anwendungen existieren in der Medizin (Diagnose), im Computerwesen (Konfiguration) und in der Erdölindustrie (Auffinden von Erdölvorkommen). Diese Anwendungen der KI sind als Expertensysteme bekannt, also als Programme, die menschliches Expertenwissen so genau und umfänglich wie möglich spiegeln und abrufbar machen. Der Vorteil ist die fast unbegrenzte Kapazität des Computergedächtnisses und seine Eigenschaft, nichts zu vergessen. Diese Vorteile können sich bei bestimmten Problemen in Nachteile verwandeln, ebenso die explosionsartig ansteigende Zahl der Kombinationsmöglichkeiten bei wachsender Zahl von Wissensmodulen.

Die Anwendungen von KI-Methoden sind jedoch weitaus vielfältiger. Die folgende unvollständige Auflistung läßt bereits auf potentielle Anwendungen in der Architektur und im Bauwesen schließen: Erkennen von Gegenständen durch Computer (Computer Vision), heute unabdingbar in Roboteranwendungen; Sprachverständnis (Language Parsing und Language Comprehension) nach Syntax und Semantik, heute eingebaut in kommerzielle Programme, auch auf Personal Computern; Suche (Search) durch gewaltige Suchräume (Search Spaces), welche mögliche Lösungen für die unterschiedlichsten Probleme beinhalten, heute Teil praktisch aller KI-Anwendungen; Logik und Deduktion, mit verschiedenen Anwendungen in

mathematischen Beweisen; Expertensysteme (Expert Systems), mit Anwendungen in Diagnose und Planung; Maschinenlernen (Machine Learning) nach verschiedenen, vom menschlichen Lernen abgeleiteten Methoden.

Architektur und KI

Keine der bisher von der KI verwendeten Methoden ist der Architektur fremd, vielmehr nutzen beide Disziplinen Erkenntnisse systematischer Forschung aus einer Vielzahl anderer Bereiche. Jedes Baugesetz ist eine Sammlung von Regeln, zusammengestelltes Expertenwissen, das als wissensbasiertes System formuliert werden kann - mit allen dazugehörigen Konflikten und Widersprüchen. Regelbasierte Systeme und Formengrammatiken sind spätestens seit Durand bekannt. Fallbasiertes Schließen ist eine bei jedem neuen Entwurf angewandte Methode, die bereits Palladio und Vitruv propagieren. Die Top-Down-Strategie findet in allen Routine-Entwurfsaufgaben und bei erfahrenen Architekten Anwendung [Akin 1986], die Bottom-Up-Strategie bei unterdefinierten Entwurfsaufgaben. Nur etwa fünf bis 15 Prozent der gesamten

Arbeitszeit verbringen Architektinnen und Architekten mit Entwerfen, der Rest der Zeit vergeht mit Akquisition, Administration, Bauüberwachung und ähnlichem. Entwerfen ist das Gebiet, in dem das Wesen des Berufs am besten zum Ausdruck kommt und das den Kreativen das meiste Vergnügen bereitet. Gerade auf dieses Gebiet konzentriert sich nun die KI-Forschung in der Architektur, bietet es doch die stärksten Anregungen und fallen hier doch die wichtigsten Entscheidungen. Die Ablehnung von KI durch Architekturschaffende war bisher besonders stark, da einerseits keine Alternative oder Verbesserung der gegenwärtigen Situation gesehen wird, andererseits eben der Entwurfs- und Planungsprozeß eine sehr von der Persönlichkeit geprägte, durch Musterbücher und Kataloge nicht ersetzbare Tätigkeit ist.

So ergibt sich die interessante Situation, daß die in der KI verwendeten Methoden Architektinnen und Architekten wohlbekannt sind, daß sie aber mit Recht spüren, daß eine KI-Implementation dieser Methoden die Arbeitsweise selbst in für sie unkontrollierbarer Weise verändern wird. Wurde der Computer bisher mit quantitativen Entscheidungen assoziiert, so erregt die Möglichkeit seiner qualitativen Fähigkeiten noch immer Besorgnis.

Die nächste Generation

Die nächste Generation der Hardware wird zunehmend von der Möglichkeit der Parallelisierung Gebrauch machen und zugleich die Geschwindigkeit der parallel geschalteten Einzelprozessoren erhöhen. Doch wird es bis zur Mitte der neunziger Jahre dauern, bis Spitzenrechner eine für das architektonische Entwerfen und Modellieren akzeptable Geschwindigkeit erreicht haben werden. Bis diese Leistung auf Personal Computern zur Verfügung steht, werden noch einmal drei bis fünf Jahre vergehen. Doch sicherlich wird sich der portable Computer weiter durchsetzen und die direkte Eingabe auf großen, farbigen Bildschirmen erlauben.

Die Visualisierungssoftware wird die neue Hardware schnell nutzen. Bei der Architektursoftware ist ein grundsätzlicher Neuanfang notwendig, der zugleich die massiv parallelen Computer nutzt und vor allem das Entwerfen als eine Reihe paralleler Prozesse simuliert. Die Rückkopplung zwischen Kontext und Entwurf wird verbessert.

In der Vernetzung stehen ebenfalls große Umwälzungen bevor. Schon jetzt wächst die Kommunikation auf den internationalen Netzwerken monatlich in zweistelligen Raten. Die in die portablen Computer eingebauten Modems werden die Kommunikation mit anderen Maschinen so einfach und so günstig wie das Telefonieren machen.

Auswirkungen der Vernetzung

Aus der Entwicklung weltweiter Kommunikationsnetze kommt eine weitere Neuerung mit weitreichenden Konsequenzen auf uns zu. Das weltweit größte Computernetzwerk, Internet, verbindet 1992 bereits mehr als 750'000 'Hosts', also Computer, die ein klei-

nes bis mittleres Netzwerk mit weiteren Computern um sich versammelt haben. Die Hosts sind miteinander mit Hochgeschwindigkeitsleitungen verbunden. Ursprünglich wurde Internet von den Regierungen verschiedener Länder finanziert und war Universitäten und offiziellen Forschern vorbehalten, aber inzwischen wächst die Zahl der Hosts fast unkontrolliert alle drei Monate um 20 bis 30 Prozent [The Economist 1992]. Daneben bestehen weitere internationale Netzwerke wie BITNET, die gemächlichere Wachstumsraten aufweisen.

Zum Vergleich: Die Übertragung einer menschlichen Stimme bedarf der Übermittlung von etwa 60'000 Bits pro Sekunde. Eine DIN A4 -Seite mit Text läßt sich durch etwa 100'000 Bits repräsentieren, eine Photographie dagegen benötigt 10'000'000 Bits oder 10 MBits. Die Übermittlung eines Films in Kino-Qualität ist erst ab Übertragungsraten pro Zeiteinheit im Giga-Bit- (GBit)- Bereich sinnvoll. Nimmt man die Größe jedes der 25 pro Sekunde zu übertragenden Bilder mit 100 MBit an, so benötigt man für die Echtzeit-Übertragung eine Verbindung, die 2,5 GBits pro Sekunde übertragen kann. Mit den zur Zeit schnellsten Verbindungen zwischen Supercomputern haben diese Glasfasernetze fast die Geschwindigkeit erreicht, mit der Daten normalerweise innerhalb eines Computers bewegt werden. Damit eröffnen sich völlig neue Möglichkeiten, denn Computer lassen sich als Konsequenz mit der entsprechenden Software zu einer Einheit zusammenfassen. Die Leistung von Supercomputern und sogar Workstations und PCs kann sich so vervielfachen. Läßt sich dieses Konzept in größerem Rahmen in die Praxis umsetzen, dann werden die Zeiten für teure Supercomputer noch schwerer. Allerdings kann eine solche Entwicklung auch zum vollständigen Zusammenbruch der Netze durch Überla-

stung führen, denn jeder wird natürlich die schnellstmögliche Simulation durchführen wollen.

Die Multimedia-Netzwerkverbindungen - zwischen ausgewählten Standorten - sind bereits Realität, und die entsprechende Software wird mit einigen neuen Workstations ausgeliefert. Die Verbreitung dieser Technik ist lediglich eine Funktion der Netzwerkgeschwindigkeit und neuer Peripheriegeräte wie Mikrophon und Videokamera.

Schnelle Netzwerke können so eine Vision wahr werden lassen, in der die Maschine, auf der man arbeitet, allenfalls noch in der lokalen Graphikleistung keine Rolle spielt. Ansonsten kann sich die Maschine je nach Bedarf vom PC zum Supercomputer mausern, indem sie über das Netz andere Prozessoren anspricht. Die Analogie zum Stromnetz wird spätestens jetzt klar: Niemand in den Industrie-Ländern,

abgesehen von Spezialfällen, wird mehr ein eigenes Kraftwerk im Haus haben; über die Steckdose beziehen wir die unterschiedlichsten Strommengen. Computer und Netzwerke gehen in dieselbe Richtung.

In Nordamerika ist nach der Aufhebung der Monopole im Kommunikationsbereich eine große Vielfalt konkurrierender Datenübermittlungssysteme entstanden. Für die Miete von Standleitungen zwischen Hosts bezahlen Europäer bis zum Zehnfachen für dieselbe Dienstleistung. Die Ausbreitung des ISDN (Integrated Services Digital Network) in Europa ist ein Hoffnungsschimmer. Doch vergleicht man die 1992 üblichen Übertragungsraten von 64 bis 128 KBit/sec mit der Aussicht auf Übertragungsraten von 2 MBit/sec, so wird klar, daß der Bezug billiger Computerleistung per Telephonanschluß noch einige Jahre in der Zukunft liegt.

Rückblick und Ausblick - eine mögliche Zukunft

In einem Aufsatz über den neuen Entwurfsprozeß und dessen Organisation schreibt eine Expertengruppe 1969 an das Britische Arbeitsministerium:

“Der Entwerfer der Zukunft wird direkte Verbindungen von seinem Büro nicht nur zu massiver Computerleistung und externen Datenbanken haben, sondern auch zu Menschen mit anderen Fähigkeiten und Interessen. Der im Computer gespeicherte Entwurf wird anderen Partnern zugänglich sein, die so an seinem Fortschritt teilnehmen. Daten für die Herstellung des Gebäudes können über diese Verbindungen direkt zu den Firmen oder sogar zur Baustelle gesandt werden. Die Entwurfszeit für einzelne Projekte wird auf einen Bruchteil ihrer jetzigen Größe schrumpfen, jedoch zugleich dem Entwerfer Gelegenheit zum detaillierten Studium von Varianten geben, für die er im Moment weder Zeit noch Gelegenheit besitzt.”
[London 1969, dt. v. Verf.]

Darin sind die Wünsche und Hoffnungen enthalten, die man traditionellerweise an CAD stellt. Die Tatsache, daß diese Forderungen fast ein Vierteljahrhundert zurückliegen und selbst in diesem schnell sich entwickelnden Gebiet noch nicht realisiert sind, spricht entweder für die Weitsicht der Verfasser oder für den Mangel an wirklichem Fortschritt. Doch muß man einfach zur Kenntnis nehmen, daß sogar bei der Erfindung eines offensichtlich nützlichen Gerätes wie des PC zwischen der Markteinführung und einer signifikanten Marktdurchdringung mindestens zehn bis 15 Jahre liegen. Die heute Allgemeingut gewordenen Hardware und Softwarekonzepte wurden vor zehn bis 15 Jahren an Universitäten und in Forschungslabors entwickelt. Es ist deshalb anzunehmen, daß viele der heute als ‘verrückte Idee’ abgewerteten Erfindungen sich später in Software und Hard-

ware implementiert wiederfinden. Gewiß gehören dazu die Konzepte der Parametrisierung, der Prototypen und der intelligenten Objekte. Entwerfen mit parametrisierten Elementen setzt sich bereits langsam durch. Prototypen erlauben die Abwandlung eines bestehenden und bewährten Konzepts an eine neue Entwurfsherausforderung in wenigen Schritten, so daß die inneren Verbindungen und die Integrität der ursprünglichen Lösung nicht verloren gehen. Intelligente Objekte schließlich werden es erlauben, auf niedrigem Abstraktionsgrad mit Objekten zu entwerfen, die über sich und ihren Kontext notwendiges Wissen besitzen und sich entsprechend verhalten.

Wir sind heute in der beneidenswerten Lage, daß die Hardwarekosten sich bei steigender Prozessorleistung ständig verringern und Fortschritte in der Netzwerktechnik die Realität eines wirklichen Daten- und Informationsverbundes näherrücken lassen. Architektinnen und Architekten können die zukünftige Entwicklung des CAD mitbestimmen, indem sie ihr Wissen, ihre Erfahrungen und Wünsche in die notwendige nächste Generation der CAD-Software einbringen. Es könnte sein, daß diese Software einen radikalen Bruch mit den bisher bestehenden Techniken vollzieht und so dem Berufsstand der Architekten in der Informationsgesellschaft das Überleben sichert. Damit wäre die Technologie, die ursprünglich als Bedrohung galt, zum Garanten der architektonischen Zukunft geworden.

Blick in das Innere eines simulierten japanischen
Hauses. Mathys, Rodriguez, Sarbach, 1991



Glossar

Aliasing		Treppenstufen-Effekt, der beim Zeichnen von Linien und Kurven auf rasterorientierten Ausgabegeräten mit begrenzter Auflösung (Bildschirme, Laserdrucker) auftritt
Ambient Light	Umgebungslicht, Streulicht	Resultat eines Beleuchtungsmodells, das eine konstante Beleuchtung auf allen Flächen erzeugt, ungeachtet ihrer Orientierung
Animation	Animation	Simulierte Bewegung eines Beobachters in einer Szene. Die Objekte in der Szene sind entweder statisch oder bewegt
Anti-Aliasing, Smoothing		Verfahren zum Reduzieren (Glätten) oder Eliminieren des Treppenstufen-Effekts
Application Program	Anwendungsprogramm, Applikation	Computerprogramm (meist kommerzieller Art) zur Lösung einer Aufgabe des Anwenders
Artificial Intelligence (AI)	Künstliche Intelligenz (KI)	Wissenschaftliche Disziplin mit zwei Zielen: 1. Simulation menschlicher Denkvorgänge wie Kreativität, Bildverstehen, Abstraktion, Sprachverstehen, Lernvermögen etc. 2. Befähigung von Computern zur Lösung von Aufgaben, die Intelligenz erfordern
ASCII (American Standard Code for Information Interchange)	ASCII-Code	Binärcode, der Zeichen, wie beispielsweise Buchstaben und Sonderzeichen, ein 7-bit-Muster zuordnet (Heute meistens auf 8-bit erweitert)
Attribute	Attribut	Beschreibende Charakteristik eines Elements
Back-Up	Sicherungskopie	Kopie einer Datei oder eines ganzen Verzeichnisses auf Diskette, Magnetband (Streamer) oder einem anderen Datenträger
Bitmap Graphics	Bitmap-Graphik	Darstellungform für Graphik, bei der das gezeigte Bild aus Rasterpunkten (als Kürzel für 'Picture Element' auch Pixel genannt) aufgebaut ist
Bitmap, Pixel Matrix	Bitmap-Darstellungen	Raster nach Position und Farbe adressierbarer Quadrate, sogenannter Pixel. Bitmap-Darstellungen sind von der Auflösung des Ausgabegeräts abhängig.
Bitplane		Videospeicher, der ein Bit Information pro Pixel speichern kann. Die Anzahl der Bitplanes bestimmt die Anzahl der darstellbaren Farben pro Pixel.
Boundary Representation		Vollständige Darstellung eines dreidimensionalen Objekts durch die Scheitelpunkte, Kanten und Flächen, die seine Oberfläche bilden
Bump Mapping		Belegung der Oberflächen eines Objekts mit höhenmodulierten Strukturen; Viel verwendet bei photorealistischen Renderings
Bus	Datenpfad	Gruppe von Leitungen, welche die Signale (Daten) zwischen verschiedenen Computerbauteilen überträgt; Der Personal Computer kennt den Daten- und Befehlsbus, den Adressbus sowie den Steuerbus.

CAAD (Computer Aided Architectural Design)	Rechnergestütztes architektonisches Entwerfen	Gesamtheit der Tätigkeiten, welche die Entwurfstätigkeit des Architekten durch die Verwendung von spezialisierten, interaktiven Informatikmitteln unterstützen und erleichtern
CAD (Computer Aided Design, Computer Aided Drafting)	Rechnergestütztes Entwerfen, Rechnergestütztes Zeichnen	Computer Aided Design: Gesamtheit der Tätigkeiten und Einrichtungen, in denen die Entwurfstätigkeit durch die Verwendung von spezialisierten, interaktiven Informatikmitteln unterstützt und erleichtert wird Computer Aided Drafting: Darstellungsinstrument, das auf einem ausschließlich graphischen Modell beruht; Aus diesem Modell lassen sich graphische oder aus der Graphik abgeleitete Automatismen entwickeln.
CD-ROM (Compact Disc - Read Only Memory)	CD-ROM	Optisches Speichermedium mit Laserabtastung, das nur gelesen werden kann; CD-ROMs können große Mengen (Größenordnung 500 Megabytes für eine 12-Zentimeter-Platte) von Daten wie Texte, digitalisierte Bilder, Sprache oder Musik speichern.
Clipboard	Zwischenablage	Temporärer Speicher innerhalb des Computers, auf dem zum Beispiel Text und Graphik abgelegt werden können
Clipping Plane	Schnitt-Ebene	Verwendung nicht sichtbarer, verschiebbarer Flächen, mit deren Hilfe sich Ausschnitte eines Modells verdecken lassen; Sinnvoll für die Herstellung von perspektivischen Schnitten und Axonometrien mit CAD-Programmen
Clock Rate	Taktfrequenz	Taktfrequenz des Prozessors, angegeben in Mhz (Millionen Zyklen pro Sekunde); Heute sind Taktfrequenzen zwischen 12 und 150 MHz üblich. Die Taktfrequenz ist einer der Indikatoren für die Arbeitsgeschwindigkeit eines Computersystems.
Color Depth	Farbtiefe	Anzahl der Bits zur Definition der Farbe eines Pixels zur Verfügung stehen; Bei einer 'Farbtiefe' von 8 Bits ergeben sich 256 verschiedene Farben.
Color Space	Farbraum	Menge der Farben, die durch das Kombinieren von bestimmten Grundfarben erzeugt werden können
Computer Graphics	Computergraphik	Technik, die der Erfassung, Bearbeitung und Darstellung von Daten in graphischer Form unter Einsatz des Computers dient
Constructive Solid Geometry (CSG)		Diese Methode wendet Boolesche Operationen (Vereinigung, Verschneidung, Differenz) auf Volumen an, um dreidimensionale Körper zu konstruieren.
Crosshair, Cursor, Pointer	Fadenkreuz, Positionsanzeiger	Frei über den Bildschirm bewegliches Symbol zur Anzeige der aktuellen Eingabeposition
Current	Aktuell	Momentan sich in Bearbeitung befindende Aufgaben; Auch: Aktive Elemente eines Modells, oder momentan aktives Dateiverzeichnis
Cyberspace		Ein globales Netzwerk als sensorisch erfahrbarer Datenraum; Der Begriff wurde von William Gibson in seinem Science Fiction Roman 'Neuromancer' geprägt.

Data Compression, Data Decompression	Datenkompression, Datendekompression	Reduktion von digitalen Input-Datenmengen mit Hilfe von Algorithmen, die das zu speichernde Datenvolumen reduzieren; Bei der Dekompression werden die Output-Daten auf den ursprünglichen Umfang expandiert.
Data	Daten	Allgemein: In beliebiger Form dargestellte Sachverhalte oder Vorgänge Speziell: Ein- und Ausgabeinformationen eines Computersystems Daten sind alphanumerische (Ziffern, Buchstaben, Sonderzeichen), graphische und sprachliche Angaben. Persistente Daten bestehen weiter nach Ablauf einer Session, temporäre Daten verschwinden nach Ablauf einer Session
Data Format	Datenformat	Logisches oder physisches Layout von Datenbeständen; Kompatible Datenformate ermöglichen das Austauschen von Dokumenten. Beispiele für gebräuchliche Formate: ASCII (Text), TARGA (Bild), TIFF (Bild), EPS (Text, Bild und Graphik), DXF (CAD)
Data Glove	Datenhandschuh	Optoelektrisches Präzisionsgerät, das wie ein Handschuh angezogen wird; Dioden, die an den Fingerspitzen angebracht sind, emittieren Infrarotlicht, das über Glasfaserkabel, die zwischen den Stofflagen liegen, ans Handgelenk geleitet und dort in elektrische Signale umgewandelt werden. Ein Prozessor wertet die Signale aus und sendet sie an den Computer, der diese Informationen weiterverarbeitet.
Default	Vorgabe	Die für Anwender eingerichtete Standard-Einstellung oder Voreinstellung der Software, die vom Hersteller als Standardlösung für eine mögliche Programm-Operation angeboten wird. In der Regel vom Anwender veränderbar
Depth Cueing	Tiefenwirkung	Algorithmus zur Darstellung von Entfernungen mittels Helligkeitsabstufungen
Directory	Dateiverzeichnis, Inhaltsverzeichnis	Dateien, die Informationen zu anderen Dateien und Dateiverzeichnissen enthalten; Ordnungsschema, um ein Dateisystem systematisch zu gliedern
Dithering		Technik, bei welcher mehrere benachbarte Pixel mit verschiedenen (aber meist ähnlichen) Farben belegt werden; Da das menschliche Auge die Farben mischt, wird die Zahl der für das Auge wahrnehmbaren Farben künstlich erhöht.
Double Buffering		Teilung des Videospeichers in zwei Hälften, wodurch die Anzahl der darstellbaren Farben halbiert, im gleichen Zug aber eine 'weichere' Animation ermöglicht wird; Während ein Bild gerechnet wird, kann ein zweites Bild dargestellt werden. Der Bildaufbau kann somit im Hintergrund durchgeführt werden.
DXF Data Format (drawing exchange format)	DXF-Datenformat	Zeichnungsaustausch-Dateiformat des CAD-Programms AutoCAD. Erlaubt einen Zeichnungsaustausch zwischen den meisten CAD-Programmen. DXF hat sich als de-facto Standard für CAD-Programme etabliert.
Editor	Verarbeitungs- programm	Programm, mit dem Texte, Tabellen, oder Modelle erstellt und geändert werden können
Entity	Element, Einheit	Objekt in einer Datenbank oder Datenbasis; Bei einigen CAD-Systemen werden die Grundelemente, aus denen eine Konstruktion im Rechner zusammengesetzt ist, als Entities bezeichnet.
Expert System (ES)	Expertensystem	Klasse von Computerprogrammen, die Fähigkeiten menschlicher Experten auf einem begrenzten, wissens- oder arbeitsintensiven Gebiet simulieren oder unterstützen; Expertensysteme verbinden einen Schlußfolgerungsmechanismus (Inference Mechanism) mit einer Wissensbasis (Knowledge-Base, KB).
Extrusion		Herausziehen eines Profils aus der Ebene, Hinzufügen der dritten Dimension
Eye Phone		Videobrille (Ein kleiner Monitor pro Auge) mit Kopfhörer und Sensoren für das Erfassen der Kopfbewegungen; Benötigt für Virtual Reality Umgebungen
File	Datei	Sammlung von Informationen, die inhaltlich zusammengehören und auf einen Massenspeicher abgelegt werden; Jeder Datei wird ein Name zugeordnet.

File Manager	Dateiverwaltung	1: Teil des Betriebssystems, der das Speichern und Wiederfinden von Dateien, sowie das Anlegen und Aktualisieren eines Inhaltsverzeichnisses auf einem Massenspeicher organisiert 2: Programm-System, das Datenfelder (z.B. Name, Vorname, Straße, Ort) in Datensätzen (z.B. alle genannten Datenfelder mit den Angaben von Herrn X) zusammenfaßt, in einer Datei (z.B. mit dem Namen 'Adressen') speichert und wiederfindet
File System	Dateisystem	Besteht aus einem Katalog (Directory) und einzelnen Files (Dateien); Der Katalog kann strukturiert sein, beispielsweise eine Hierarchie von Katalogen darstellen. In UNIX sind die Kataloge hierarchisch organisiert.
Flat Shading, Facet Shading		Darstellung mit Facetten, wobei jede Facette (Polygonfläche zwischen den Verbindungen des Drahtmodells) eine separate Farbe einnehmen, jedoch keine Facette in mehr als eine Farbe unterteilt werden kann
Fractal	Fraktal	Objekt, das mit einem rekursiven Algorithmus generiert werden kann; Fraktale sind durch Selbstähnlichkeit gekennzeichnet, wobei sich das Motiv in immer kleinerem Maßstab fortlaufend wiederholt.
Frame Buffer		Speicher zur Aufbewahrung von Bildern über einen kurzen, bestimmten Zeitraum
Gouraud Shading, Smooth Shading	Gouraud Schattierung	Weiterentwicklung des Flat Shading; Beim Schattieren der Polygonfläche wird für jede Ecke des Polygons eine Farbe errechnet, die Farben für die dazwischenliegenden Pixel werden interpoliert. Dadurch wird der beim Flat Shading störende Eindruck einer aus geraden Einzelflächen bestehenden Oberfläche gemindert.
Hardware	Hardware	Oberbegriff für alle mechanischen, magnetischen, elektrischen und elektronischen Komponenten eines Computersystems
Hidden Line, Hidden Surface	Verdeckte Kanten, Verdeckte Flächen	Wegrechnen der in einer bestimmten Projektion unsichtbaren Kanten, beziehungsweise Flächen eines dreidimensionalen Körpers; Es gibt eine Vielzahl von Algorithmen, welche diese Aufgabe mehr oder weniger effizient lösen. Beispiele: Warnock-, Watkins-, Z-Buffer-Algorithmen
Hypermedia		Hypertext-Dokumente, die zusätzlich multimediale Informationen wie Ton (Sprache, Klang), Text, Photos und Video-Animationen enthalten; Multimediale Informationen sind vernetzt (es ist z.B. möglich, von einem Text zu einer Graphik oder einem Klang und umgekehrt zu gelangen) und interaktiv. Anwendungsmöglichkeiten: On-line Handbücher, Unterrichtsprogramme, elektronische Enzyklopädien, Präsentation für Werbezwecke
Hypertext		Medium zur Informationsspeicherung, in dem zwischen beliebigem Text und anderen Objekten durch Verbindungen (Links) Beziehungen definiert werden können
Icon	Ikone, Pictogramm	Kleines Symbol auf dem Bildschirm, das Applikationen, Dateien oder Verzeichnisse darstellt; Sie werden meist mit der Maus ausgewählt.
Image Mapping		Bei diesem Verfahren wird ein flaches Bild auf ein meist dreidimensionales Objekt projiziert.
Image Processing	Digitale Bildverarbeitung	Softwaregestützte Methode für die digitale Bearbeitung von Bildern
Input	Dateneingabe	Jede Information, die (auch über die Peripherie) die Zentraleinheit des Computers erreicht
Input Device, Input Unit	Eingabegerät, Eingabeeinheit	Peripheriegeräte zur Eingabe kodierter Daten wie Tastatur, Digitizer, Lichtstift oder Mikrofon

Interface	Schnittstelle	<p>1. Hardware-Schnittstelle: Anschlußmöglichkeit zwischen Computern und Peripheriegeräten. Eine elektronische Schaltung, die Geräte oder Bausteine einander anpaßt</p> <p>2. Software-Schnittstelle: Anpassungsprogramm, zum Beispiel ein In- und Output-Programm; Spezifiziert werden in der Regel Anzahl, Reihenfolge, Art und zeitliche Abfolge von Daten und Parametern.</p>
Lambert Cosine Shading		Einfache, facettenartige Schattierung (Flat Shading)
LAN (Local Area Network)	LAN	Computernetzwerk in einem lokal begrenzten Bereich; Ein LAN arbeitet mit hohen Übertragungsraten, womit verschiedenartigste Informationen ausgetauscht werden können.
Layer	Zeichnungsebene, Schicht	Eine Zeichnung kann auf verschiedene Zeichnungsebenen verteilt werden. Eine Zeichnungsebene, beziehungsweise Layer, kann beispielsweise nur das Mauerwerk, ein anderer technische Installationen enthalten. Übereinandergelagert ergeben sich dann im Beispiel das Mauerwerk mit den technischen Installationen.
Link	Verbindung, Verknüpfung	<p>Physisch: Verbindung von Computern oder Computer-Teilen untereinander</p> <p>Symbolisch: Durch einen Link können Daten in ein anderes Dokument eingefügt werden, ohne eine Kopie zu erstellen. Somit erscheinen auch Änderungen an dem eingefügten Dokument automatisch im neuen Dokument.</p>
Macro	Makro	Zusammenfassung von Anweisungen zu einer ansprechbaren Einheit, um häufig durchgeführte Arbeitsgänge zu automatisieren und damit zu vereinfachen
Mainframe	Mainframe	Eine der Bezeichnungen für EDV-Großanlagen.
Menu	Menü	Auf dem Bildschirm graphisch angezeigte Übersicht aktuell wählbarer Programmfunktionen
Micro Computer	Mikro-Computer	Computer basierend auf Microchips, ursprünglich in der Größenordnung eines Personal Computers (PC); Der Begriff umfaßt aber auch ganz spezielle Computersysteme: Meß- und Steuereinheiten, Bordcomputer etc.
Modeling	Modellierung	Beschreibung der Charakteristiken eines Objekts wie Form, Oberfläche und Bewegung, sowie der zugehörigen Manipulationsmöglichkeiten mit dem Computer
Modem (Modulator-Demodulator)	Modem	Gerät für die Datenübertragung, welches die digitalen Signale eines Computers in analoge Impulse für das öffentliche Telephonnetz und zurück verwandelt. Die Übertragungsgeschwindigkeit wird in bit pro Sekunde (bps) beschrieben.
Multimedia		Computerunterstützte Informationssysteme, die verschiedene Medien (Text, Bild, Ton und Animationen) einsetzen; Die Informationen sind interaktiv. Der Mensch nimmt die Informationen nicht wie bei Fernsehen oder Video einfach auf, sondern kann eingreifen und sie seinen Bedürfnissen gemäß aktiv nutzen. Im Gegensatz zu Hypermedia sind hier die verschiedenen Medien nicht verknüpft.
Multitasking System	Mehrprogramm-Betrieb	Betriebsart des Computers, bei welcher mehrere Aufgaben oder Programme gleichzeitig ablaufen; Der Computer verarbeitet scheinbar mehrere Aufgaben (Tasks) gleichzeitig, indem er die Pausen nützt, in denen die Zentraleinheit nicht beschäftigt ist. Beispiel: Eine aufwendige Tabellenkalkulation oder ein Sortiervorgang in einer Datenbank laufen im Hintergrund ab, während man gleichzeitig im Vordergrund einen Text schreiben kann.
Multiuuser System	Mehrfachbenutzer-Betriebssystem	Computersystem, zu dem mehrere Anwender gleichzeitig Zugriff haben

Network	Netzwerk	Verkettung von Computersystemen zu Kommunikations- oder Rechenzwecken
Network Protocol	Netzwerkprotokoll	Sammlung von 'Verkehrsregeln', die darüber bestimmen, wie Daten von den verbundenen Geräten gesendet und empfangen werden können; Das Protokoll ist unabhängig vom verwendeten Verkabelungssystem.
Object Oriented Methods	Objektorientierte Methoden	(Programmier-) Methoden, die Probleme nach Objekten strukturieren (im Gegensatz zur Strukturierung nach Funktionen); Die Kommunikation zwischen Objekten geschieht über Botschaften (Messages).
Operating System	Betriebssystem	Sammelbegriff für die Gesamtheit der Systemprogramme, die den gesamten Arbeitsablauf eines Computers steuern und überwachen
Output	Datenausgabe	Von einem Computer an ein externes Gerät, wie zum Beispiel Drucker, Plotter, Modem, Video, gesendete Daten; Auch: interner Output an andere Funktionen
Output Device	Ausgabegerät	Peripheriegerät zur Darstellung der Computerergebnisse wie Bildschirm, Drucker, Plotter
Password	Kennwort	Schutzeinrichtung gegen unberechtigten Zugriff; Nur wer das richtige Kennwort eingibt, kann die entsprechend geschützten Operationen durchführen.
Path	Pfad	Liste, welche die Namen von Dateiverzeichnissen, möglicherweise den Namen einer Datei enthält; Um eine Datei innerhalb des Filesystems anzusprechen, muß als Adresse der Pfadname angegeben werden.
Periphery Devices	Peripheriegeräte	Geräte, die einem oder mehreren Computern zugeordnet sind, zum Beispiel Drucker, Plotter, Scanner
Permission	Zugriffsrechte	In Mehrfachbenutzer-Umgebungen Kennzeichnung der Operationstypen, die für eine Datei ausgeführt werden können: Lesen (Read), Schreiben (write), Ausführen (execute)
Phong Shading	Phong-Schattierung	Die Farbe eines jeden Pixels der Fläche wird aus seiner relativen Lage zu den Lichtquellen der Szene separat berechnet
Pixel (Picture Element)	Bildpunkt	Ein Pixel stellt einen Punkt des Bildes dar, insbesondere einen Punkt des Bildes auf dem Bildschirm.
Plotter	Zeichenmaschine	Automatisches, computergesteuertes Zeichengerät. Die heute meistbenutzten Plotter sind Stift-, Thermo-, Inkjet-, Laser- und elektrostatische Plotter.
PostScript	PostScript	Seitenbeschreibungssprache der Firma Adobe Systems; Komplette, kurvenorientierte Programmiersprache, die Informationen über den Aufbau und das Aussehen einer ganzen Seite an den Drucker zur Weiterverarbeitung weitergibt
Printer	Drucker	Peripheriegerät zur Ausgabe von Rechnerinformationen auf Papier; Über Drucker werden vorwiegend alphanumerische Daten und graphische Daten in gerasteter Form ausgegeben. Man unterscheidet unterschiedliche Druckertypen wie Matrixdrucker, Laserdrucker, Tintenstrahldrucker, Thermo- drucker und Sublimationsdrucker.
Printer Plotter		Printer mit graphischen Fähigkeiten
Processor	Prozessor	Elektronischer Baustein, der ein Computersystem oder Teile davon nach vorgegebenen Programmen steuern kann; Daneben gibt es Prozessoren für besondere Aufgaben, zum Beispiel Arithmetik- und Graphikprozessoren.
Radiosity	Radiosity	1984 in Cornell entwickelte Methode; Von jeder Stelle einer Oberfläche werden Stärke und Richtung des ausgestrahlten Lichts berechnet. Die Stärke des Lichts setzt sich dabei aus dem vom Körper emittierten und dem reflektierten Licht zusammen. Gegenüber dem Ray Tracing sind die Berechnungen hier zwar aufwendiger, benötigen aber bei anschließenden Änderungen des Bildes,

		bei Bewegungsabläufen (Animation) oder beim Verschieben des Blickwinkels keine umständlichen Neuberechnungen. Aus diesem Grund eignet sich dieses Verfahren gut für Animationen.
Ray Tracing	Strahlverfolgung	1979 in den Bell Laboratories von Turner Whitted entwickelte Methode; Die Lichtstrahlen werden nicht von den Körperoberflächen in Richtung des Sehenden, sondern in umgekehrter Richtung verfolgt und berechnet. Vorteil: Es müssen nur diejenigen Strahlen verfolgt beziehungsweise berechnet werden, die auch wirklich vom Auge erfaßt werden. Das Ergebnis sind äußerst realistische Bilder. Bei beweglichen Bildern, insbesondere beim Verschieben des Beobachtungspunktes, müssen die Berechnungen für jedes Bild neu durchgeführt werden, was zu erheblichen Rechenzeiten führt.
Record	Datensatz	Logisch zusammengehörende Dateneinheit, etwa eine Briefadresse; Die Länge eines Datensatzes ist nicht fixiert. Sie wird je nach dem zu beschreibenden Objekt verschieden ausfallen. Eine Datei kann aus solchen Datensätzen aufgebaut werden.
Rendering	Photorealistische Darstellung	Herstellungsprozeß von Bildern unter Berücksichtigung von Farben, Beleuchtung (Beleuchtungsmodellen wie Ray-Tracing etc.), Durchsichtigkeit, Oberflächeneigenschaften (eventuell mit Image- und Texture Mapping)
RGB	RGB	Farbmodell mit den Grundfarben rot, grün und blau, die additiv kombiniert werden
Ruled Surface	Regelfläche	Fläche, die durch Verschieben einer Kurve entlang einer Leitgeraden beschrieben wird
Shading	Schattierung	Farbschattierung der Partien eines Objekts entsprechend einer oder mehrerer angenommener Lichtquellen
Shell		Ein Programm unter UNIX, das der Kommunikation des Benutzers mit dem Betriebssystem dient
Snap	Fang	Bei dieser Option bewegt sich der Cursor auf einem vorher definierten Fangaster, oder auf besonderen Punkten (wie Endpunkt und Mittelpunkt) von Elementen. Somit kann man ohne absolute Koordinatenangaben zeichnen.
Software	Software	Sammelbezeichnung für alle Programme, also alle nicht-maschinellen Bestandteile eines Computers
Solid	Volumen	Der Körper wird als Volumen definiert, so daß jede einzelne Körperstelle (auch das Innere) beschrieben ist.
Solid Model	Volumenmodell	Das Modell wird im Rechner mit Hilfe von Solids (Volumen) dargestellt
Solid Modeling	Volumenmodellieren	Operieren mit volumetrischen Grundelementen. Festkörper können unter anderem verschmolzen und zerschnitten werden (Boolesche Operationen).
Solid Texturing		Neben der Oberfläche werden auch die Schnittflächen mit der entsprechenden Textur versehen. Grundlage der Texturierung ist hier eine dreidimensionale Textur.
Specular Reflection		Reflexion von direktem Licht, das auf eine glänzende Oberfläche fällt; Das Ergebnis sind Glanzlichter auf dem Objekt.
Spline		Mit Hilfe von mehreren Stützpunkten wird ein Kurvenzug mit Krümmungen konstruiert. Grundlage ist eine nicht-lineare Funktion, deren Verlauf durch Stützpunkte und Spannungsvektoren bestimmt wird.
Subdirectory	Unterverzeichnis	In einer Verzeichnishierarchie eine Ebene tiefer liegendes Directory
Surface Modeling	Oberflächenmodellieren	Körper werden durch ihre Begrenzungsflächen (Oberflächen) definiert; Diese Oberflächen können ausgefüllt dargestellt werden. Das Innere des Körpers

		bleibt jedoch undefiniert. Bei einem Schnitt durch den Körper wird man deshalb nur die Schnittekanten der Oberflächen mit der Schnittebene sehen.
Text File	Textdatei	Datei, in der die Information in Form von Textzeichen gespeichert ist
Texture	Textur	Bestimmtes Muster, das auf eine Fläche gelegt wird (u.U. auch repetitiv); Beispiele: Textildgewebe, Textilfaserung, Gestein- und Holzstrukturen, prozedurale Texturen; Texturen können auch oberflächenmodulierend wirken.
Texture Mapping		Digitale Projektion beliebiger Texturen auf dreidimensionale, auch gekrümmte Oberflächen.
Trim	Stutzen, Trimmen	Treffen oder überschneiden sich zwei Linien, so werden die entsprechenden Enden durch Trimmen zueinander verlängert, bzw. abgeschnitten. Die Linienenden bilden dann eine Ecke.
Transparency	Durchsichtigkeit, Transparenz	Das Maß, mit dem einfallendes Licht ein Objekt durchdringen kann. Das Gegenteil von Opacity
UNIX	UNIX	Standard-Betriebssystem für Mehrfachbenutzer- und Mehr-Programmbetrieb; Es wurde 1969 an den AT&T Bell Laboratories entwickelt. Zur Zeit existieren verschiedene UNIX-Versionen.
User Interface	Benutzeroberfläche	Oberbegriff für alle Arten von Bedienungsschnittstellen zwischen den Kommandos eines Betriebssystems, einer Applikation und dem Benutzer
User Interface, Graphical	Graphische Benutzeroberfläche	Meist mit Maus steuerbare Bedienungsschnittstelle am Computer, die auf graphischen Symbolen (Ikonen), Menüs, sowie beweglichen Fenstern besteht, in denen verschiedene Programme ablaufen können; Häufige Anwendungen: Datei-Verwaltung, Aufrufen und Bedienen von Applikationen, zum Teil auch Programmierhilfe.
Utility	Dienstprogramm	Hilfsprogramm zur Lösung von Problemen, die bei verschiedenen Anwendungen gleich sind; Dienstprogramme reichen von Datenverwaltungs-Systemen, Bildschirmformatgestaltungen bis zu System- und Wartungsfunktionen.
Virtual Reality (VR)	Virtual Reality	Die Möglichkeit, mit Hilfe der Computertechnik, speziellen Eingabegeräten und Stereobrille als Sichtgerät einen virtuellen, nicht real existierenden Raum frei erfahren zu können. Die Hauptmerkmale sind die Immersion (das Gefühl, sich wirklich im zu untersuchenden Raum zu befinden) und die Interaktivität.
Wireframe	Drahtmodell, Gittermodell, Kantenmodell	Flächen und Körper werden durch die sie begrenzenden Linien dargestellt. Bei einem Schnitt durch den Körper wird man deshalb nur die Schnittpunkte der Linien mit der Schnittebene sehen.
Workstation	Arbeitsstation	Leistungsfähiges Computersystem; Eine Workstation muß unter anderem folgende Minimal-Anforderungen erfüllen: - 32-64 Bit Architektur für Prozessor und Datenpfad - Multitasking Betriebssystem für simultane Bearbeitung mehrerer Programme (meistens wird UNIX verwendet) - hohe Graphikleistung, Auflösung von 1024x1280 Bildschirmpunkten, Multi-Windowing - Netzwerkfähigkeit (z.B. Ethernet, Token Ring)
Z-Buffer		Speichereinheit, in der die Z-Koordinaten (also in Projektionsrichtung) der dargestellten Objekte abgelegt werden
Z-Buffering		Technik zur Unterdrückung verdeckter Linien. Für jedes Pixel wird die Z-Koordinate (Entfernung zum Betrachter) gespeichert. Dargestellt werden nur die Objekte, die sich vor allen anderen befinden.
Zoom	Zoom	Vergrößern (Zoomfaktor >1) oder verkleinern (Zoomfaktor <1) eines Bildes

Bibliographie

- Akin 1982 Akin, Ömer und E. F. Weinel, Representation and Architecture, Silver Spring, Maryland (Information Dynamics) 1982
- Akin 1986 Akin, Ömer, Psychology of Architectural Design, London (Pion) 1986
- Alexander 1964 Alexander, Christopher, Notes on the Synthesis of Form, Cambridge, Massachusetts (Harvard University Press) 1964
- Alexander 1966 Alexander, Christopher, A City is Not a Tree, in: Design, Vol. 206, Febr. 1966, S. 46-55
- Alexander 1977 Alexander, Christopher, S. Ishikawa, und M. Silverstein, mit M. Jacobson, I. Fiksdahl-King und S. Angel, A Pattern Language, New York (Oxford University Press) 1977
- Baumeister 1947 Baumeister, Willi, Das Unbekannte in der Kunst, Stuttgart (Schwab Verlag) 1947
- Behaneck 1991 Behaneck, Marian, Dieter J. Heimlich und Peter Wossnig, Vom CAAD zum Bild, Neustadt a.d. Weinstrasse (SOFT-TECH Eigenverlag) 1991
- Boegli 1984 Boegli, Hans, Aventicum, die Römerstadt und das Museum, in: Archäologischer Führer der Schweiz 20, (Association Pro Aventico) 1984
- Borning 1981 Borning, A., The Programming Language Aspects of ThingLab. A Constraint-Oriented Simulation Laboratory, in: ACM Transactions on Programming Languages and Systems, Vol. 3 (4) Okt. 1981
- Brooks 1985 Brooks, Rodney, Programming in Common Lisp, New York (Wiley & Sons) 1985
- Burns 1993 Burns, Howard, The future role of computers in the GSD, in: Zusammenfassung, Computer-Workshop, Graduate School of Design, Harvard University, 8. Mai 1993
- Carlsen 1991 Carlsen, Niels Vejrup, Modelling User Interface Software, PhD Dissertation, Department of Graphical Communication, Technical University of Denmark, Lyngby 1991, S. 6
- Chen 1990 Chen, Chen Cheng und Schmitt, Gerhard, Design as Learning, in: Proceedings of the Second European Conference on Application of Artificial Intelligence and Robotics to Building Architecture and Civil Engineering, University of Liège, Belgien, März 1990, Paris (Hermès) 1990
- Chen 1991 Chen, Chen Cheng, Analogical and Inductive Reasoning in Architectural Design Computation, Dissertation, Abteilung für Architektur, ETH Zürich Mai 1991
- Churchland 1990 Churchland, Paul M. und Churchland, Patricia Smith, Ist eine denkende Maschine möglich?, in: Spektrum der Wissenschaft, März 1990, S. 47-54
- Clancey 1986 Clancey, William J., Heuristic Classification, in: Kowalik, Janusz (Hrsg.), Knowledge Based Problem Solving, New Jersey, New York (Prentice-Hall) 1986

-
- Cooper 1992 Cooper, Douglas, The Visible City, Video, 22 Minuten,
Carnegie Mellon University, Pittsburgh, Pennsylvania 1992
- Cousin 1992 Cousin, Jean-Pierre, La Station Valode et Pistre,
in: d'Architectures, Nr. 30, Nov. 1992, S. 54-59
- Cox 1987 Cox, Brad, Object Oriented Programming. An Evolutionary Approach,
Reading, Massachusetts (Addison-Wesley) 1987
- Coyne 1990 Coyne, Richard, M. A. Rosenman, A. D. Radford, M. Balachandran und
J. S. Gero, Knowledge-Based Design Systems,
Reading, Massachusetts (Addison- Wesley) 1990
- Danahy 1992 Danahy, John, The Computer-Aided Studio Critic: Gaining Control of What We
Look at, in: G. Schmitt (Hrsg.),
CAAD futures '91, Wiesbaden (Vieweg-Verlag) 1992, S. 121-138
- Darke 1984 Darke, J., The Primary Generator and the Design Process, in: N. Cross (Hrsg.),
Developments in Design Methodology,
Chichester (John Wiley & Sons) 1984
- Dave 1992a Dave, Bharat und G. Schmitt, Information Systems for Urban Analysis and
Design Development, in: Proceedings of Design and Decision
Support Conference '92, TU Eindhoven, Juli 1992
- Dave 1992b Dave, Bharat und Gerhard Schmitt, CAAD Programmentwicklung, Skript Win-
tersemester 1992/93,
Professur für Architektur und CAAD, ETH Zürich 1992
- Dave 1993 Dave, Bharat, A Computer Assisted Diagramming System, Dissertation,
Abteilung für Architektur, ETH Zürich, Febr. 1993
- Degler 1993 Degler, Hans-Dieter, Wissenszwerge unter Druck - Die Krise auf dem Weg ins
Informationszeitalter,
Der Spiegel, Nr. 14, 1993, S. 150-158
- Dette 1992 Dette, Klaus (Hrsg.), PC-Einsatz in der Hochschulausbildung, Berlin
(Springer-Verlag) 1992
- Doczi 1981 Doczi, György, The Power of Limits. Proportional Harmonies in Nature, Art and
Architecture,
Boulder & London (Shambhala) 1981
- Dreyfus 1987 Dreyfus, Hubert L. und Dreyfus, Stuart E., Von den Grenzen der Denkmaschine
und dem Wert der Intuition,
Hamburg (Rowohlt) 1987
- Evans 1963 Evans, T.G., A Heuristic Program to solve Geometric-Analogy Problems,
Ph.D. Thesis, Department of Mathematics,
MIT, Cambridge, Massachusetts 1963
- ETH 1993 ETH Jahresbericht '92, Schulleitung der ETH Zürich,
Presse- und Informationsdienst der ETH Zürich, Zürich, 1993

-
- Fenves 1990 Fenves, S. J., U. Flemming, C. Hendrickson, M. L. Maher und G. Schmitt, Integrated Software Environment for Building Design and Construction, in: Computer Aided Design, Vol. 22, Nr. 1, Jan. 1990, S. 27-36
- Flemming 1981 Flemming, Ulrich, The Secret of the Casa Giuliani Frigerio, in: Environment and Planning B, 8 1981, S. 87-96
- Flemming 1986 Flemming, Ulrich, The Role of Shape Grammars in the Analysis and Creation of Designs, in: Y. Kalay (Hrsg.), Computability of Design, New York (John Wiley & Sons) 1986, S. 245-272
- Flemming 1987 Flemming, Ulrich, More than the Sum of Parts: The Grammar of Queen Anne Houses, in: Environment and Planning B, 14 1987, S. 323-350
- Flemming 1988a Flemming, Ulrich, Rule-Based Systems in Computer Aided Architectural Design, in: M. Rychener (Hrsg.), Expert Systems for Engineering Design, New York (Academic Press) 1988, S. 93-112
- Flemming 1988b Flemming, Ulrich, Coyne, R., Glavin, T., und Rychener, M., A Generative Expert System for the Design of Building Layouts - Version 2, in: Gero, John (Hrsg.) Artificial Intelligence in Engineering Design, New York (Elsevier, Computational Mechanics Publications) 1988
- Galle 1992 Galle, Per, Mac Alex 1.0, A System for Architectural Sketching, in: Technical Report PG 92-a, Department of Graphical Communication, Technical University of Denmark, Lyngby, 1992
- Gandelsonas 1991 Gandelsonas, Mario (Hrsg.), The Urban Text, Cambridge, Massachusetts (The MIT Press) 1991
- Garrett 1986 Garrett, J. H. und S. Fenves, A Knowledge-based Standards Processor for Structural Component Design, in: Technical Report R-85-157, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, Sept. 1986
- Gero 1988 Gero, John, M. L. Maher und W. G. Zhang, Chunking Structural Design Knowledge as Prototypes, The Architectural Computing Unit, Department of Architectural Science, University of Sydney, Sydney 1988
- Gero 1990 Gero, John, Design Prototypes: A Knowledge Representation Schema for Design, in: AI Magazine, Vol. 11, Nr. 4 1990, S. 26-36
- Gero 1993 Gero, John, und M. L. Maher (Hrsg.), Modeling Creativity and Knowledge-Based Creative Design, (Lawrence Earlbaum) 1993
- Grabow 1983 Grabow, Stephen, Christopher Alexander. The Search for a New Paradigm in Architecture, Boston, Massachusetts (Oriel Press) 1983
- Hammond 1989 Hammond, Kristian, Case-Based Planning. Viewing Planning as a Memory Task, Boston (Academic Press) 1989
- Hancock 1986 Hancock, Les und Morris Krieger, The C Primer, New York (McGraw Hill) 1986
- Harrington 1983 Harrington, Steven, Computer Graphics, New York (McGraw-Hill) 1983
- Hendrickson 1987 Hendrickson, C. T., C. A. Zozaya-Gorostiza, D. Rehak, E. Baracco-Miller und P. Lim, An Expert System for Construction Planning,

-
- in: ASCE Journal of Computing in Civil Engineering, American Society of Civil Engineers, Nr. 113 (5), Okt. 1987
- Henze 1992 Henze, H., C. Welzel, A. Schäfer, M. Bendig, H. Gerken und K. M. Baumgarten, Software für den Architekturbereich, IAP Institut für Architektur und Planungstheorie, Universität Hannover, 1992
- Heylliard 1991 Heylliard, Caroline (Hrsg.), Valode & Pistre 1990, Paris (Agence Minimum) 1991
- Heylliard 1992 Heylliard, Caroline (Hrsg.), Valode & Pistre 1991, Paris (Agence Minimum) 1992
- Hillier 1984 Hillier, W., J. Musgrove und P. O'Sullivan, Knowledge and Design, in: N. Cross (Hrsg.), Developments in Design Methodology, Chichester (John Wiley & Sons) 1984
- Holland 1986 Holland, J. H., K. F. Holyoak, R. E. Nisbett und P. R. Thagard, Induction, Cambridge, Massachusetts (The MIT Press) 1986
- Hua 1992 Hua, Kefeng, Ian Smith, Boi Faltings, Shen-Guan Shih und Gerhard Schmitt, Adaptation of Spatial Design Cases, in: John Gero (Hrsg.), Artificial Intelligence in Design '92, Dordrecht (Kluwer Academic Publishers) 1992, S. 559-575
- Jacobs 1991 Jacobs, Stephen Paul, The CAD Design Studio, New York (McGraw-Hill) 1991
- Jones 1963 Jones, J. C. und D. G. Thornley (Hrsg.), Conference on Design Methods, Oxford (Pergamon) 1963
- Kahlen 1989 Kahlen, Hans, CAD Einsatz in der Architektur, Stuttgart (Kohlhammer) 1989
- Keane 1988 Keane, Mark T., Analogical Problem Solving, New York (Ellis Horwood Limited) 1988
- Klotz 1989 Klotz, Heinrich (Hrsg.), Jahrbuch für Architektur 1989, Braunschweig/Wiesbaden (Vieweg-Verlag) 1989
- Knight 1992 Knight, Terry, Designing with Grammars, in: G. Schmitt (Hrsg.), CAAD futures '91, Wiesbaden (Vieweg-Verlag) 1992, S. 33-48
- Knudsen 1992 Knudsen, S. E., und A. J. Schai, Gedanken zu Massiv-Parallel-Rechnern an der ETHZ, in: Informatik Bulletin, ETH Zürich Nr. 70, Juli 1992, S. 4-8
- Koning 1981 Koning, H. und J. Eizenberg, The Language of the Prairie: Frank Lloyd Wright's Prairie Houses, in: Environment and Planning B, 8 1981, S. 295-323
- Kramel 1992 Kramel, Herbert und C. C. Chen, BAU: A Knowledge-Based System for the Investigation of a Basic Architectural Unit, in: G. Schmitt, (Hrsg.), CAAD futures '91, Wiesbaden (Vieweg-Verlag) 1992, S. 329-346
- Küng 1993 Küng, M. und Th. Glatthard, EDV in der Raumplanung, in: Vermessung, Photogrammetrie, Kulturtechnik, Jan. 1993, S. 55-56.
- Lacy 1991 Lacy, Bill, 100 Contemporary Architects, New York (Harry N. Abrams) 1991
- Langley 1987 Langley, P., Machine Learning and Grammar Induction, in: Machine Learning, 2 1987, S. 5-8

-
- Larkin 1987 Larkin, J. und H. Simon, Why a Diagram is (Sometimes) Worth Ten Thousand Words, in: Cognitive Science, Vol. 11 1987, S. 65-99
- Laseau 1980 Laseau, Paul, Graphical Thinking for Architects and Designers, New York (Van Nostrand Reinhold) 1980
- Le Corbusier 1963 Le Corbusier, Ausblick auf eine Architektur, in: Bauwelt Fundamente, Bd. 2 Berlin (Ullstein-Verlag) 1963
- Lenart 1991 Lenart, Mihaly, Expertensysteme in der Architektur und im Bauwesen, Basel (Birkhäuser-Verlag) 1991
- London 1969 Ministry of Public Building and Works, Directorate of Research and Information, Computer-Aided Architectural Design, (Crown Copyright), London 1969, S. 69
- Madrazo 1992a Madrazo, Leandro, Design as Formal Language, in: Proceedings, ECAADE '92 International Conference, Barcelona, Nov. 1992, S. 319-329
- Madrazo 1992b Madrazo, Leandro, Design Education with Computers, in: G. Schmitt (Hrsg.), CAAD futures '91, Wiesbaden (Vieweg-Verlag) 1992, S. 77-96
- Madrazo 1993a Madrazo, Leandro, From Sketches to Computer Images: A Strategy for the Application of Computers in Architectural Design, in: Proceedings, IMARA '93, Monte Carlo, Febr. 1993
- Madrazo 1993b Madrazo, Leandro, 3 Objects: Object-object, Object-building, Object-city, in: Proceedings, International Symposium on Creativity and Cognition, Loughborough University, April 1993
- Maher 1987 Maher, M. L. und P. Longinos, Development of an Expert System Shell for Engineering Design, in: International Journal of Applied Engineering Education, Oxford (Pergamon Press) 1987
- Mandelbrot 1982 Mandelbrot, Benoit B., The Fractal Geometry of Nature, San Francisco (W. H. Freeman and Co) 1982
- Mäntylä 1988 Mäntylä, Martti, An Introduction to Solid Modeling, Rockville, Maryland (Computer Science Press) 1988
- March 1984 March, Lionel, The Logic of Design, in: N. Cross (Hrsg.), Developments in Design Methodology, Chichester (John Wiley & Sons) 1984
- McDermott 1987 McDermott, Drew und Eugene Charniak, Introduction to Artificial Intelligence, Reading, Massachusetts (Addison Wesley) 1987, S. 6
- McKim 1980 McKim, Robert, Experiences in Visual Thinking, Monterey, California (Brooks/Cole) 1980
- Meißner 1992 Meißner, U., P. Mitschke-Collande und G. Nitsche (Hrsg.), CAD im Bauwesen, Berlin (Springer-Verlag) 1992
- Michalski 1986 Michalski, Ryszard S., Understanding the Nature of Learning: Issues and Research Directions, in: Michalski, R. S., Carbonell, J. G. und Mitchell, T. M. (Hrsg.), Machine Learning. An Artificial Intelligence Approach, Vol. II, Tioga, California (Tioga Publishing Company) 1986

-
- Mitchell 1977 Mitchell, William, Computer-Aided Architectural Design, New York (Van Nostrand Reinhold) 1977
- Mitchell 1978 Mitchell, Tom M., Version Space: An Approach to Concept Learning, Ph.D. Thesis, Computer Science Department, Stanford University 1978
- Mitchell 1987 Mitchell, William J., R. Liggett und T. Kvan, The Art of Computer Graphics Programming, New York (Van Nostrand Reinhold) 1987
- Mitchell 1990a Mitchell, William J., The Logic of Architecture, Cambridge, Massachusetts (The MIT Press) 1990
- Mitchell 1990b Mitchell, William J., R. S. Liggett und M. Tan, Top-Down Knowledge-Based Design, in: M. McCullough (Hrsg.), The Electronic Design Studio, Cambridge, Massachusetts (The MIT Press) 1990, S. 137-148
- Mitchell 1990c Mitchell, William J., The Uses of Inconsistency in Design, in: Y. Kalay, (Hrsg.), Evaluating and Predicting Design Performance, New York (John Wiley & Sons) 1990
- Mitchell 1992a Mitchell, William J., Integrating Shape Grammars and Design Analysis, in: G. Schmitt (Hrsg.), CAAD futures '91, Wiesbaden (Vieweg-Verlag) 1992, S. 17-32
- Mitchell 1992b Mitchell, William J., The Reconfigured Eye - Visual Truth in the Post-Photographic Era, Cambridge, Massachusetts (The MIT Press) 1992
- Neitzke 1992 Neitzke, Peter und Carl Steckeweh, Centrum - Jahrbuch Architektur und Stadt 1992, Braunschweig/Wiesbaden (Vieweg-Verlag) 1992
- Newell 1972 Newell, Alan und Herbert Simon, Human Problem Solving, New Jersey (Prentice-Hall) 1972
- Norman 1992 Norman, Donald A., The Psychology of Everyday Things, New York (Basic Books Inc., Publishers) 1988, S. 178
- Pardey 1992 Pardey, Hans-Heinrich, Computer Kids, in: Frankfurter Allgemeine Magazin, 6, November 1992, S. 24
- Peitgen 1986 Peitgen, H. O. und P. H. Richter, The Beauty of Fractals. Images of Complex Dynamical Systems, Berlin (Springer-Verlag) 1986
- Peitgen 1988 Peitgen, H. O. und D. Saupe, The Science of Fractal Images, Berlin (Springer-Verlag) 1988
- Pevsner 1976 Pevsner, Nikolaus, A History of Building Types, New York (Princeton University Press) 1976
- Pu 1990 Pu, Pearl, und Boi Faltings, The Limits of Logic in Reasoning about Compositional Systems, Laboratoire d'Intelligence Artificielle, Département d'Informatique, EPF Lausanne, Febr. 1990
- Radford 1988 Radford, A. D. und J. S. Gero, Design Optimization in Architecture, Building and Construction, New York (Van Nostrand Reinhold) 1988
- Reiser 1992 Reiser, M. und Niklaus Wirth, Programming in Oberon: Steps beyond Pascal and Modula, New York (ACM Press) 1992

-
- Roozenburg 1991 Roozenburg, N. F. M. und N. G. Cross, Models of the Design Process: Integrating Across the Disciplines, in: Design Studies, Vol. 12, Nr. 4, Okt. 1991, S. 215-220
- Rudofsky 1964 Rudofsky, Bernard, Architecture without Architects, a Short Introduction to Non-Pedigreed Architecture, Garden City, New York (Doubleday & Company) 1964
- Rychener 1988 Rychener, Michael D. (Hrsg.), Expert Systems in Engineering Design, Boston, Massachusetts (Academic Press) 1988
- Saggio 1992 Saggio, Antonino, Object Based Modeling and Concept-Testing, in: K. Kensek und D. Noble (Hrsg.), Mission, Method, Madness, ACADIA Workshop Proceedings, Charleston, North Carolina 1992
- Schaad 1988 Schaad, Isolde, Schmetterlingsmuster der Mathematik - Die Liebe zur Geometrie - Urs B. Roth, du, Okt. 1988, S. 38-42
- Schank 1982 Schank, Roger C., Dynamic Memory: A Theory of Learning in Computers and People, Cambridge, Massachusetts (Cambridge University Press) 1982
- Schmitt 1983 Schmitt, Gerhard, Auswirkungen des Energieproblems auf die Architektur - unter besonderer Berücksichtigung von Computersimulationen und Computer Aided Design als Entscheidungshilfen im Entwurfsprozeß, Dissertation, Technische Universität München 1983
- Schmitt 1988a Schmitt, Gerhard, Microcomputer Aided Architectural Design, New York (John Wiley & Sons) 1988
- Schmitt 1988b Schmitt, Gerhard, IBDE, VIKa, ARCHPLAN: Architectures for Design Knowledge Representation, Acquisition und Application, in: Yoshikawa H. und Holden T. (Hrsg.), Intelligent CAD II, Proceedings of the Second IFIP WG 5.2 Workshop on Intelligent CAD, Cambridge, England (Elsevier Science Publishers, North-Holland) Sept. 1988
- Schmitt 1989a Schmitt, Gerhard, C. C. Chen und I. Paoli, Expert Systems In Architecture, in: Fourth Semi-Annual Progress Report to IBM, Department of Architecture, Carnegie Mellon University Pittsburgh, Pennsylvania Jan. 1989
- Schmitt 1989b Schmitt, Gerhard, Architectural Pre-Processor to Engineering Expert Systems, in: Proceedings of the IABSE Colloquium 1989 on Expert Systems in Civil Engineering, Bergamo Okt. 1989
- Schmitt 1989c Schmitt, Gerhard, CAAD - Script Wintersemester 1989/90, Professur für Architektur und CAAD, ETH Zürich 1989
- Schmitt 1991 Schmitt, Gerhard und C. C. Chen, Third Semi-Annual Research Report to IBM on Graphical Knowledge Acquisition, Abteilung für Architektur, ETH Zürich 1991
- Schmitt 1992a Schmitt, Gerhard, Architektur und Künstliche Intelligenz, in: Peter Neitzke und C. Steckeweh (Hrsg.), Centrum. - Jahrbuch Architektur und Stadt 1992, Wiesbaden (Vieweg-Verlag) 1992, S. 221-231
- Schmitt 1992b Schmitt, Gerhard (Hrsg.), CAAD futures '91, Wiesbaden (Vieweg-Verlag) 1992
- Searle 1990 Searle, John R., Ist der menschliche Geist ein Computerprogramm? in: Spektrum der Wissenschaft, März 1990, S. 40-47

-
- Shih 1993 Shih, Shen-Guan und Gerhard Schmitt, The Use of Post Interpretation for Grammar-Based Generative Systems, in: Proceedings, International Workshop on Formal Design Methods for CAD, Talinn, Estland, Juni 1993
- Simon 1982 Simon, Herbert A., The Science of the Artificial, Cambridge, Massachusetts (The MIT Press) 1982
- Simon 1992 Simon, Herbert A., People and Computers: Their Roles in Creative Design, in: Keynote Lecture, Second International Conference on Artificial Intelligence in Design, Carnegie Mellon University, Pittsburgh 22. Juni 1992
- Solomonoff 1959 Solomonoff, R., A New Method for Discovering the Grammars of Phrase Structure Languages, in: Proceedings, International Conference on Information Processing, 1959
- Stenvert 92 Stenvert, Ronald, Constructing the Past: Computer-Assisted Architectural-Historical Research, Eigenpublikation, Utrecht 1992
- Stiny 1978 Stiny, George und W. J. Mitchell, The Palladian Grammar, in: Environment and Planning B, 5 1978, S. 5-18
- Stiny 1980 Stiny, George, Introduction to Shape and Shape Grammars, in: Environment and Planning B, 7 1980, S. 343-351
- Sutherland 1963 Sutherland, I., Sketchpad, A Man-Machine Graphical Communication System, in: Proceedings 1963 Spring Joint Computer Conference AFIPS, Vol. 23 1963
- The Economist 1992 The fruitful, tangled trees of knowledge, in: The Economist, 20. Juni 1992, S. 99-102
- Tennent 1982 Tennent R., Grundlagen der Programmiersprachen, (Hanser-Verlag) 1982
- Thalmann 1991 Thalmann, Daniel, New trends in Visualization and Animation, New York (Springer-Verlag) 1991
- v. Maur 1985 v. Maur, Karin (Hrsg.), Vom Klang der Bilder. Die Musik in der Kunst des 20. Jahrhunderts, München (Prestel-Verlag) 1985
- VDI 1973 VDI Richtlinie 2222 Blatt 1 (Entwurf), Konzipieren technischer Produkte, Düsseldorf (VDI-Verlag) 1973
- Weiss 1973 Weiss, Richard, Häuser und Landschaften der Schweiz, Erlenbach-Zürich (Eugen Rentsch-Verlag) 1973
- Wiesmann 1990 Wiesmann, Stefan, Anforderungen für produktorientiertes CAD in der Architektur und Konzepte für deren Lösung, in: Vortrag am 4. CIP-Status Kongreß, TU Berlin, Okt. 1990
- Winograd 1983 Winograd, T., Language as a Cognitive Process, Reading, Massachusetts (Addison-Wesley) 1983
- Wirth 1985 Wirth, Niklaus, Programmieren in Modula-2, Berlin (Springer-Verlag) 1985
- Wirth 1988 Wirth, Niklaus, The Programming Language Oberon, Software Practice and Experience, 18, Juli 1988, S. 671-690
- Wolstencroft 1989 Wolstencroft, John, Restructuring, Reminding and Repair: What's Missing from Models of Analogy, in: AICOM, Vol. 2, Nr. 2 1989
- Zechner 1992 Zechner, Christoph, Architekturhacker im virtuellen Raum, ACS-Preis '92, Architektenkammer Hessen, Wiesbaden 1992, S. 38

Stichwortverzeichnis

- Abfragefunktionen, 179
- abstrakte Form, 165
- abstraktes Gebäudemodell, 54
- Abstraktion und Modellbildung, 38
- Abstraktion, 56
- Abstraktionsgrad, 38
- ACS-Preises, 80
- Adaptation, 48
- Adaptierung bestehender Objekte, 184
- Akin, 44
- Alexander, Christopher, 26, 30, 68
- Algorithmen, 33
- Alhambra, 142
- Alphabet graphischer Zeichen, 56
- Analogien von Entwurf und Sprache, 105, 112
- Analphabeten, 205
- Analyse, 129
- Analyse-Synthese-Modell, 27
- Analysephase, 28
- Animation, 81, 100, 129f, 130, 158, 219
- Anpassung des Lehrstoffs, 86
- Anschaffung einer CAD-Anlage, 213
- Anwendermodell, 209
- Anwendung kommerzieller Programme, 86
- AppleTalk (Mac OS), 207
- Arbeitsmodelle, 99
- Architectural Space Laboratory, 128
- Architektonische Abstraktionen, 100
- Architektur als Wissenschaft, 24
- Architektur und KI, 223
- Architektur und Maschine, 84
- Architekturausbildung, 90, 95, 132
- Architekturbüro, 204, 211
- Architekturgeschichte, 48
- Architekturlehrer, 25
- Architekturschulen, 140
- Archivierung, 90
- ARCHPLAN, 95
- Assemblersprache, 60
- Audio, 221
- Ausführungsphase, 40
- Austauschen von Alternativen, 74
- automatische Wissensaneignung, 50
- Avenches, 120
- Aventicum, 120
- Axonometrie, 108

- Bailey, Simon, 184
- Beaux-Arts, 68, 76
- Bedürfnisse der Anwender, 209
- Benutzeroberflächen, 38, 81, 132, 204
- Benutzerverhalten, 101
- Berufstand der Architekten, 226
- Betriebssysteme, 207
- Bibliothek von Typen, 74
- Bildkompression, 68
- Bildverarbeitungsmethoden, 158
- Blackboard, 98
- Bofill, Ricardo, 141
- Boolesche Operationen, 57, 72
- Bottom-Up-Methode, 27, 44
- Breitensuche (Breadth First), 62
- Büroautomatisierung, 84
- Bürokommunikation, 218

- C++, 64
- C, 60
- CAAD futures'91, 150
- CAAD Komponenten und Programme, 216
- CAAD, 8, 32, 177
- CAD-Erfahrung, 214
- CAD-Programme, 205, 208, 211

-
- CAD-Umgebung, 216
 - CAD-Zeichner, 212
 - Cardo, 42, 120
 - Carnegie Mellon University, 25, 95
 - Case-Based Reasoning, 184
 - CDT, 56
 - Chen, Cheng-Chen, 50f, 51
 - Cigogniertempel, 120
 - CISC (Complex Instruction Set Computing), 211
 - Civitas Helvetiorum, 120
 - Client-Server Umgebung, 140
 - Compound Objects, 107, 112
 - Computer Vision, 222
 - Computer, 9, 206
 - Computer-Instrumente, 52
 - Computerinfrastruktur, 87, 140, 204
 - Computerisierung bestehender Methoden, 93
 - Computermodell, 99, 209, 212,
 - Computerrepräsentation, 31, 35
 - Computersprachen (Computer Languages), 208, 211
 - Computerumgebung, 204
 - Constraints, 191
 - Cooper, Douglas, 78
 - CORE, 95
 - CPU, 206
 - Cut-and-Paste-Methoden, 91

 - Danahy, John, 181
 - Danteum, 128
 - Datenaustauschformate, 213
 - Datenbank, 42, 113, 120, 122, 134, 178, 211, 217,
 - Datenbank-Management-Systeme (DBMS), 92, 217
 - Datenbankabfragen, 180
 - Datenbankfunktionalität, 179
 - Datenbankstrukturen, 179
 - Datenmodelle, 38

 - Datenraum, 168
 - Datenrepräsentation, 179
 - Datensicherheit, 181
 - Datenübermittlungssysteme, 225
 - Dave, Bharat, 33, 55f, 56, 63, 133, 178, 179ff, 180, 181, 182
 - Decumanus, 42, 120
 - Deduktion, 222
 - Design as Language, 104
 - Design Methods, 30
 - Design Methods, Bewegung, 29
 - Designfokus (Logical Zoom), 53, 78
 - Designprozeß, 27
 - Desktop Paradigm, 209
 - Desktop Publishing (DTP), 219
 - Detailierungsgrade, 53, 74, 76, 110, 105,
 - Diagnose, 223
 - Diagramme, 52, 56
 - digitale DIA-Aufnahmen, 158
 - digitale Geometrie, 33
 - digitaler Schnittplatz, 221
 - digitales Geländemodell, 126, 180
 - Digitalisierung, 122
 - Dimensionalitäts-Reduktion, 184, 185
 - Dimensionskonflikte, 184
 - Disketten (Floppy Disk), 206
 - DOS, 207
 - Drachenstele, 162
 - Drahtmodelle (Wireframes), 57
 - dreidimensionales Modell, 143
 - drucksensitive Tablett, 219
 - Echtzeit-Visualisierung, 180
 - EDESYN, 97
 - Einkapselung von Informationen, 64
 - Einkapselung von Wissen in Klassen, 62
 - Einsetzungspunkt, 74

-
- Einzelbilder (Frames), 220
Eisenmann, Peter, 68
Eizenberg, Julie, 44
elektronischer Bleistift, 93
Energiesimulation, 54
Energieverbrauch, 40, 92, 218
Engeli, Maia, 71, 90, 119, 122
Entscheidungsfindung, 95
Entwerfen, 31
Entwicklungsabschätzung, 204
Entwurf einer Bibliothek, 114
Entwurf und Sprache, 104
Entwurfshilfen, 88, 93f
Entwurfsinstrument, 74
Entwurfsmethodik, 26
Entwurfsprozeß, 25, 28, 144
Entwurfsraum, 105
Entwurfsumgebung, 30, 178, 180, 191
Entwurfsunterricht, 25
Erdölindustrie, 222
Ersetzen eines Objektes, 74
Erzeugung neuer Prototypen, 46
Espace & Stratégie, 141
Ethernet, 207
Expertensysteme (Expert Systems), 62, 222f
Expertenwissen, 48
- Fall (Case), 184
Fall-Anpassung (Case Adaptation), 184
Fall-Datenbank (Case Base), 48, 173, 189
Fallbasiertes Schließen (Case-Based Reasoning), 48, 94, 184
fallbasiertes Entwerfen (Case-Based Design), 184
Faltings, Boi, 184
FDDI, 207
Feederle, 130
Fenves, Steven, 95
Festplatte (Hard Disk), 206
Fleischli, 135f
Flemming, Ulrich, 66, 97, 188
FOOTER, 98
Formalisierung des Entwurfsprozesses, 26, 30, 32
Formengrammatiken, 44, 52, 66, 68, 105, 120, 123, 134, 223
Fort- und Weiterbildung, 214
Fortran, 60
fraktale Denkweisen, 68
fraktale Dimension, 68
fraktale Geometrie, 68
Fraktale, 68, 120, 123, 134
Frames, 101
Froebelsche Bauklötze, 66
funktionale Constraints, 185
Funktionen und Kontrollstrukturen, 132
- Gaudi, Antoni, 76
Gebäudedarstellung, 40
Gebäudeform, 187
Gebäudemanagement, 92
Gebäudematerialien, 54
Gebäudetypen, 72, 187
gedankliche Werkzeuge, 36
Generator-Annahme-Analyse-Modell, 28
Generieren und Testen, 44
Geo-Informationssysteme (GIS), 179
Geodatenverarbeitung, 178
Geometrie, 52, 57
Geometrie-Editoren, 57
Geometriemodell, 38, 57, 95, 127
geometrische Alternativen, 74
geometrische Komplexität, 169
geometrische Transformation, 66
Gesamtkomposition, 106
Gesamtlösung, 42, 44
Gesichtsmodelle, 70
Gigabytes, 206
GKS, 211
Glasfasernetze, 224
Gouraud, 211
Grammatik, 188
grammatische Regeln, 66
Graph, 33, 184f
Graphik, 211
Graphikbibliotheken, 211

-
- Graphikbranche, 101
 - graphische Benutzeroberfläche, 118
 - graphische Regel, 66
 - Gray, Eileen, 175
 - Grohe, 31, 36f
 - Großprojektion, 127
 - Grün, Armin, 178
 - Grundlagenforschung, 222
 - GUI (Graphical User Interface Systems), 210

 - haptische Qualitäten, 100
 - Hardware, 181, 205, 213, 216
 - Hardwareanforderungen, 216
 - Hauptspeicher, 206
 - Hecker, Stefan 175ff
 - heuristische Regeln, 222
 - Hierarchie, 64, 112
 - hierarchische Komposition, 42
 - hierarchische Objekte, 112
 - hierarchische Strukturen, 129
 - hierarchisches Denken, 74
 - Hirschberg, Urs, 93
 - Historismus, 76
 - höhere Programmiersprachen, 60
 - HOOPS, 211
 - Host, 224
 - Hua, Kefeng, 184

 - Induktion, 29
 - Information Processing Theory, 28
 - Informationsgesellschaft, 226
 - Informationstiefe, 76
 - Informationsverarbeitung, 30
 - Infrastruktur, 85, 205
 - Ingenieurwissenschaften, 26
 - Inkonsistenzen, 91
 - Instrumentarium, 89
 - Insulae, 122
 - Integration, 212, 214
 - Integrationsmöglichkeit, 217
 - integrierte Programme, 213, 218
 - integrierten Gebäude-Entwurfsumgebung, 95
 - integrierter Entwurf, 95, 98
 - integriertes System, 216
 - intellektuelles Instrumentarium, 211
 - intelligente CAAD-Systeme, 184
 - intelligente Cursor, 192
 - intelligente Entwurfsumgebung, 88
 - intelligente Gebäude, 76
 - intelligente Objekte, 191
 - intelligente Objekte, 226
 - intelligente Prototypen, 124
 - interne Schulung, 214
 - Internet, 224
 - IPX (Novell), 207
 - ISDN, 141, 225
 - italienischer Rationalismus, 128
 - Iteration und Rekursion, 132

 - Kämpfer, Thomas, 93, 126ff
 - Kassettenroboter, 206
 - KI, 222
 - Kit-of-Parts, 122
 - Klassifikation, 26
 - Knight, Terry, 44
 - kognitive Modelle, 172, 174
 - kombinatorische Explosion, 66
 - Kommunikation Mensch-Maschine, 36
 - Kommunikation, 146, 207f, 224
 - Kompatibilität der Repräsentation, 34
 - Kompatibilität, 212f
 - Komplexität, 166
 - Komposition, 106
 - Kompositionsregeln, 56
 - Konfiguration, 222
 - Koning, Hank, 44, 66
 - Konstruktion im Raum, 108
 - Konstruktion, 187
 - Konstruktionserleichterung, 217
 - Konstruktionsmaterial, 185
 - Konstruktionsprozeß, 217
 - Konstruktionsvereinfachungen, 191

-
- Kontextmodell, 114
Konventionen, 33
Konzept, 33, 166
Korrekturkosten, 40
Kostenschätzungen, 54
Kramel, Herbert, 51, 66, 89
Kreativität, 212
Kreativitätssteigerung, 84
Künstler, 168
Künstliche Intelligenz, 35, 40, 191, 204
Kurmann, David, 61, 65
Kurzzeitgedächtnis, 28, 206
- Langzeitgedächtnis, 28, 206
Le Corbusier, 118
Left Hand Side, LHS, 66
Lehre und Machina, 85
Lehrinhalte, 85, 87
Lenart, Mihali, 62
Lernen durch Analogie, 50
Lernen durch Deduktion, 50
Lernen durch Induktion, 50
Lernen, 222
Lerninhalte, 85
Lernmethoden, 85
Lerup, Lars, 118
Libeskind, Daniel, 68
Lichtausbreitung, 99
Lichtsimulation, 54
Liebknecht-Luxemburg-Denkmal, 68
Lineamenta-CAAD, 150
lineares und dynamisches Programmieren, 54
linguistisches Modell, 109, 129
Lisp, 60, 132
LocalTalk, 207
Logik, 222
Luebke, Chris, 118f
- Macintosh-Betriebssystem (Mac OS), 207
Madrazo, Leandro 73ff, 104ff, 142, 150, 152ff
Magnetbänder, 206
- Maher, Mary Lou, 97
Mainframes, 206
Maison en bord de mer, 175
Mandelbrot, 68
Mario Campi, 93, 126, 146
Maschinengeneration, 206
Maschinenlernen (Machine Learning), 50, 223
Maschinensprache, 60
Massenmodell-Programm, 114
Massenspeicher, 206
massiv parallele Computer, 208, 223
Mathys, 226
Matter, Hans Uli, 110
Maximumsuche (Hill Climbing), 44, 62
Medium Papier, 99, 218
Megabytes, 206
Meister, Martina, 180ff
Mental Image, 174
Methoden, 36, 64, 191
MFLOPS, 206
Mies van der Rohe, 68
Minicomputer, 206
MIPS, 206
Mitchell, William, 72
Modelle und Präsentation, 99
Modelle, 38, 211, 223
Modellieren mit Prototypen, 118
Modelliermethode, 78
Modellierungsumgebung, 37, 72, 130
Modellüberfliegungen, 100
Modems, 141
Modulor, 54
Moneo, 30
Motif, 207
Müller, Christian, 175
Multi-Tasking, 207
Multi-User, 207
Multimedia, 181, 224f
Multiple Repräsentation, 110
- natürliche Sprachen, 208

-
- Netzwerke, 204
 - Netzwerksoftware, 208
 - neuronale Netze (Neural Network), 222
 - Newell, Alan, 28
 - Norman, Donald, 210
 - Notebook, 219
 - Nouvel, Jean, 141

 - Oberflächenmodelle (Surface Models), 57
 - Objekt-Detaillierung, 76
 - Objektattribute, 105
 - Objektklasse, 64
 - objektorientierte Datenbanken, 180, 217
 - objektorientierte Programmierung (OOP), 52, 64
 - Objektorientiertes Entwerfen, 114
 - Open GL, 211
 - Openlook, 207
 - Operating System, 207
 - Operationen (Operations), 57, 64
 - optische Datenträger (CD, Compact Disc), 206
 - Ordnung und Chaos, 91
 - Organisationsformen, 91
 - Organisationsprobleme, 91
 - Oswald, Franz, 178

 - Paint-System, 52
 - Papierdokumente, 218
 - papierloses Büro, 218
 - Parallelisierung, 223
 - Parameter, 70, 123, 163
 - Parametrisierte Formen, 134
 - parametrisierte Programme, 172
 - Parametrisierung, 52, 70, 120, 226
 - Pattern Language, 30
 - PC, 205, 216, 226
 - Peitgen, Otto, 68
 - Penbook, 219
 - Peripherie, 209
 - PHIGS, 211
 - Phong, 211
 - Photogrammetrie, 178
 - Photorealistische Darstellungen, 40, 99f, 129
 - physische Modell, 38
 - PLANEX, 98
 - Planung, 223
 - Planungsprozeß, 40
 - Platzhalter, 74, 143
 - Polytrim, 180
 - Prädikate und Funktionen, 132
 - Präsentationsmedium, 101
 - Präsentationsmodelle, 99
 - Präsentationszeichnungen, 212
 - Praxis und Machina, 140
 - Praxis, 204
 - Primitive, 72
 - Problemlösung, 33
 - Produktionen (Productions), 62, 212
 - Produktionssysteme, 44, 66
 - Produktivitätsgewinne, 214
 - Programme (Software), 208
 - Programme für die Konstruktion, 216
 - Programmentwicklung, 132
 - Programmiersprachen, 60, 134
 - Prolog, 60
 - Proportionen, 54, 187
 - Prototyp-Anpassung, 46
 - Prototyp-Verfeinerung, 46
 - Prototypen, 46, 72, 94, 120, 226
 - Prozessoren, 211

 - Rahmen (Frames), 33, 62, 101
 - RAM, 206
 - rationale Entwurfsmodelle, 30
 - Raumsynthesizer, 81
 - Realität, 40, 81
 - Rechenleistung, 206
 - Refvem, Sharon, 31, 71, 146ff, 160f, 184
 - regelbasiertes System, 172
 - Regeln (Rules), 33, 62, 208
 - Rekonstruktion, 42, 125, 128f, 175
 - rekursive Kombination, 44
 - rekursive Simulation, 173

-
- Relation, 33
Relationale Datenbanken, 211, 217
Relationale Prototypen, 124
Rendering, 100, 177, 211, 219
Repräsentation von Architektur, 32
Restaurierung, 175
Restriktionen (Constraints), 54
Restriktions-basiertes Modellieren, 64
Right Hand Side, RHS, 66
RISC (Reduced Instruction Set Computing), 211
Roboteranwendungen, 222
Rodriguez, 226
Rossi, Aldo 118
Rotations-Parameter, 106
Roth, Urs, 162ff
Rückwärtsdeduktion (Backward Chaining), 62, 184
Rudofsky, Bernard, 24
Rütimann, Christoph, 168, 171
Rychener, Michael, 62
- Saggio, Antonino, 128, 148
Sanierung, 175
Sarbach, Amadeo, 226
Saurer, Karl, 158
Schattierungsalgorithmen, 211
Schemata (Schema), 62
Schichten (Layer), 78
Schreibtischparadigma, 209
Semantik, 35, 191, 222
Semantische Netze, 33
Shih, Shen-Guan, 67, 69, 71, 97, 184, 188
Simon, Herbert, 28, 30, 42
Simulation, 40, 87, 146, 160
Simulationsprogramme, 52
Single-Tasking, 207
Single-User-System, 207
Sladoljev, Zoran, 94, 116, 148
- Smalltalk, 64
Smith, Ian, 184
Softwarebibliotheken, 211
SPEX, 97
Spiegelung, 106
Sprachverständnis, 222
Spurensicherung, 176
SQL, 179
Städtebau, 178
Standard Software, 140
standardisierte Bauteile, 70
STANLAY, 97
Sterling, 118
Stiny, George, 66
strukturierte Abfragesprachen, 217
STRYPES, 97
Substitution, 52, 47, 108, 109, 126
Subtraktion, 57
Suche (Search), 222
Suchmechanismen, 40
Suchräume (Search Spaces), 28, 222
Suchstrategien, 44
Sulzer, 126
Sumi und Burkhalter, 160
Summerauer, Veronika 137ff
Supercomputer, 224
Sutherland, Evan, 209, 222
Symbole, 74, 208
symbolische Darstellung, 78
syntaktische Aspekte des Entwurfs, 114
Syntax, 9, 32, 35, 191, 222
System-Umgebung, 134
- Tabellenkalkulationsprogramm, 54, 84, 218
TCP/IP (UNIX), 207
Technologieabschätzung, 204
Terragni, Giuseppe, 128

-
- Textverarbeitung, 218
 - Textverarbeitungsprogramme, 84, 218
 - Thalmann, Nadia und Daniel, 219
 - Themenraster, 178, 180
 - ThingLab, 64
 - Tiefensuche (Depth First), 62
 - Top-Down-Methode, 27, 42, 78
 - Topographie, 147
 - Topologie, 70, 123
 - Traditionelle Architekturmodelle, 99
 - Traditionelle Entwurfsprozesse, 25
 - Traditionelle Programme, 52, 60
 - Transformationen, 57, 185
 - Typen und Variationen, 52, 72, 105, 144

 - UIDE (User Interface Design Environments), 210
 - UIMS (User Interface Management Systems), 210
 - UIS (User Interface Systems), 210
 - Umfeld (Context), 62
 - Ungers, Simon, 68
 - UNIX, 207, 221
 - unscharfe Logik (Fuzzy Logic), 222
 - Unterhalt von Netzwerken, 208

 - Valode et Pistre, 140
 - van der Mark, Eric, 59, 101, 121, 123, 124, 125, 135ff, 144ff, 151, 155, 157ff, 190
 - Variationen (Instances), 62, 74, 106, 126
 - Vererbung, 64
 - Verfeinerung eines Entwurfsansatzes, 78
 - vernetztes Animationssystem, 221
 - vernetztes CAD-System, 221
 - vernetztes Multimedia-System, 221
 - vernetztes Renderingsystem, 221
 - Vernetzung-Networks, 207, 213
 - Verschieben, 106
 - Verschneidungsbildung, 57

 - Verwaltung von Dateien, 92
 - Vezin, Claude, 148f
 - Video, 100, 127, 160, 221
 - Videokamera, 221, 224
 - Virtual Reality (VR), 40, 81, 205, 221
 - virtuelle Architektur, 81
 - virtuelle Modelle, 169
 - Visualisierung, 134, 176f
 - Visualisierungsprogramme, 99, 155, 223
 - Vokabular, 105f
 - Volumenmodelle (Solid Models), 57, 89, 129
 - von Lucius, Daniel, 103
 - von Neumann-Maschine, 208
 - Vorwärtsdeduktion (Forward Chaining), 62, 184

 - Weiss, Richard, 188
 - Wenz, Florian, 81, 101, 148, 168
 - Windows, 207
 - Wire Frame Darstellung, 80
 - wirklichkeitsnahe Darstellung, 100
 - Wirth, Niklaus, 208
 - Wissensbasierte Programme, 44, 52, 62
 - Wissenschaftliche Entwurfsprozesse, 26
 - wissenschaftliche Forschung, 24
 - wissenschaftlichen Rekonstruktion, 176
 - Wissensingenieur, 66
 - Wissensverarbeitung, 35
 - Workstation, 216, 224

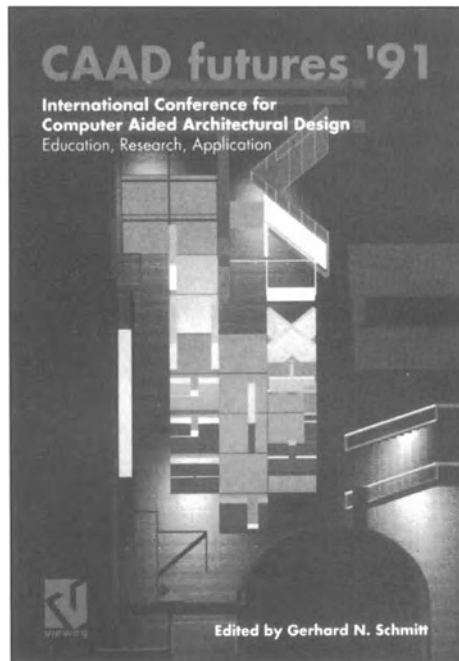
 - Zahlen- und Texteditoren, 54
 - Zeichenprogramm, 89
 - Zeichnen, 168
 - Zeidler, Annelies, 122
 - Zielvorstellung (Goal State), 62
 - Zimmermann, 128f, 131
 - Zugriffsrecht, 207
 - zusammengesetztes Objekt (Compound Object), 112

CAAD futures '91

Computer Aided Architectural Design Futures
Education, Research, Applications

edited by Gerhard N. Schmitt

1992. 596 pages. Hardcover
ISBN 3-528-08821-4



CAAD ist mehr als ein Werkzeug, es erweitert den architektonischen Horizont. Die vierte internationale CAAD-Konferenz (Zürich, 1.-3. Juli 1991) beschäftigte sich mit Fragen der Ausbildung, mit den Perspektiven des computerunterstützten architektonischen Entwerfens ebenso wie mit Problemen der praktischen Anwendung. 35 der weltweit wichtigsten Arbeiten auf den Gebieten Ausbildung, Forschung und Anwendung wurden durch Lehrer, Praktiker und Forscher selbst vorgestellt: William Mitchell (Harvard), Charles Eastman (University of California, Los Angeles), Leandro Madrazo (ETH Zürich) Gianfranco Carrara (La Sapienza, Rom), John Gero (University of Sydney), Yehuda Kalay (State University of New York), Shen-Guan Shih (ETH Zürich), Robert Johnson (University of Michigan), Edna Shaviv (Technion, Haifa), Daniel Thalmann (EPF Lausanne), Patricia Alkhoven (Utrecht), um nur einige Namen zu nennen.

Die Publikation dokumentiert die Referate der gesamten Konferenz. Sie informiert über den Stand der Entwicklung von CAAD im internationalen Maßstab ebenso wie über die Perspektiven einer Disziplin, die mit Grund als visionäres Gestaltungsmittel bezeichnet werden kann.