



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Document

D-95-09

**PROXIMA:
Ein System zur Generierung graphischer
Abstraktionen**

Antonio Krüger

September 1995

**Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH**

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: + 49 (631) 205-3211
Fax: + 49 (631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: + 49 (681) 302-5252
Fax: + 49 (681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, Sema Group, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct systems with technical knowledge and common sense which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Computer Linguistics
- Programming Systems
- Deduction and Multiagent Systems
- Document Analysis and Office Automation.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Dr. Dr. D. Ruland
Director

PROXIMA:
Ein System zur Generierung graphischer Abstraktionen

Antonio Krüger

DFKI-D-95-09

This work has been supported by a grant from The Federal Ministry of Education, Science, Research and Technology (FKZ ITWM-9400).

© Deutsches Forschungszentrum für Künstliche Intelligenz 1995

This work may not be copied or reproduced in whole or part for any commercial purpose. Permission to copy in whole or part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of the Deutsche Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Federal Republic of Germany; an acknowledgement of the authors and individual contributors to the work; all applicable portions of this copyright notice. Copying, reproducing, or republishing for any other purpose shall require a licence with payment of fee to Deutsches Forschungszentrum für Künstliche Intelligenz.

PROXIMA:
Ein System zur Generierung graphischer
Abstraktionen

Antonio Krüger

12. September 1995

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziel und Gegenstand der Arbeit	5
1.2	Aufbau der Arbeit	5
2	Graphische Abstraktionen	7
2.1	Begriffserklärung	7
2.2	Abstraktionsziele	9
2.3	Abstraktionstypen	11
2.4	Einfluß von Randbedingungen auf die Wahl von Abstraktions- typen	15
2.5	Zusammenfassung	17
3	Relevante Vorarbeiten	18
3.1	Relevante Arbeiten aus Psychologie und Graphik-Design . . .	18
3.1.1	Begriffspsychologie	18
3.1.1.1	Die klassische Theorie	19
3.1.1.2	Die Gradiententheorie	22
3.1.1.3	Diskussion	23
3.1.2	Design und Gestalttheorie	24
3.1.2.1	Diskussion	26
3.2	Relevante Arbeiten aus dem Bereich der Informatik	27
3.2.1	Abstraktionen im System APEX	27
3.2.1.1	Diskussion	34
3.2.2	Ein System zur Generierung von Skizzen	36
3.2.2.1	Diskussion	38
3.2.3	Ein System zur Konvertierung von CAD-Modellen in FEM-Modelle	38

3.2.3.1	Diskussion	39
3.2.4	Ein System zur funktional basierten Objekterkennung	39
3.2.4.1	Diskussion	41
3.2.5	Ein System zur „Verschönerung“ von Graphiken	41
3.2.5.1	Diskussion	43
4	Konzepte eines Systems zur rechnergestützten Abstraktion	44
4.1	Die Formulierung der Abstraktionsaufgabe	45
4.1.1	Darstellungsklassen	46
4.1.2	Signifikante Attribute	48
4.2	Repräsentation der graphischen Elemente	49
4.2.1	Externe Objektrepräsentation	50
4.2.2	Interne Objektrepräsentation	51
4.3	Der Abstraktionsvorgang	51
4.4	Forderungen an ein Abstraktionssystem	52
5	PROXIMA: Ein Werkzeug zur Erzeugung graphischer Abstraktionen	54
5.1	Geometrische Wissensstrukturen	55
5.1.1	Objektrepräsentationen	55
5.1.1.1	Externe Objektrepräsentation	55
5.1.1.2	Interne Objektrepräsentation	56
5.1.2	Signifikante Attribute	57
5.2	Abstraktionsziele und Abstraktionstypen	57
5.3	Die Formulierung der Abstraktionsaufgabe	60
5.4	Der Abstraktionsvorgang	61
5.4.1	Der lokale Kontext	62
5.4.2	Die Modellabstraktion	65
5.4.2.1	Berechnung der räumlichen Relationen	65
5.4.2.2	Parametrisierung der Abstraktionstypen	67
5.4.2.3	Durchführung der Abstraktion	69
5.4.3	Die Projektionsabstraktion	74
5.4.3.1	Parametrisierung der Abstraktionstypen	74
6	Arbeiten mit PROXIMA an Anwendungsbeispielen	76
6.1	Die Beispieldomäne	76
6.2	Betriebsmodi	76

6.2.1	Interaktiver Betriebsmodus	77
6.2.2	Vollautomatischer Betriebsmodus	78
6.2.3	Halbautomatischer Betriebsmodus	78
6.3	Einsatzfelder	79
6.3.1	Subkomponente eines Intellimediasystems	79
6.3.2	Interaktive Generierung von 3D-Illustrationen	83
6.3.2.1	Das System TOPAS	84
6.3.2.2	Das System AWI	85
6.3.3	Animationen	88
6.3.4	Zusammenarbeit mit einem begrifflichen Referenzierungssystem	89
6.3.5	Graphische Suchsysteme	90
6.3.6	Konvertierung von CAD-Modellen in FEM-Modelle	92
7	Die Implementation von PROXIMA	93
7.1	Repräsentation der geometrischen Modelle	93
7.1.1	Objektrepräsentationen	94
7.1.2	Datenstrukturen für signifikanten Attribute	97
7.2	Regelinterpretier	97
7.3	Abstraktionsalgorithmen	98
7.3.1	Modellabstraktion	98
7.3.2	Projektionsabstraktion	101
7.4	Testumgebung	102
7.4.1	Ladeanweisung	102
7.4.2	Fensterlayout	103
7.4.3	Die Menüs	103
7.4.3.1	Aktionsmenü	103
7.4.3.2	2D-Abstraktion	104
7.4.3.3	3D-Abstraktion	104
7.4.3.4	Zielmodus	105
7.5	Programmierschnittstelle	105
7.5.1	Globale Variablen	105
7.5.2	Initialisierung der Abstraktionshierarchie	106
7.5.3	Interaktiver Betriebsmodus	106
7.5.4	Vollautomatischer Betriebsmodus	107

8	Zusammenfassung und Ausblick	109
8.1	Erzielte Ergebnisse	109
8.2	Erweiterungsvorschläge	110
8.2.1	Verfeinerung von Abstraktionszielen und Rand- bedingungen	110
8.2.2	Weitere Abstraktionstypen	110
8.2.3	Automatische Evaluierung generierter Abstraktionen .	111
8.2.4	Abstraktionsregeln	111
8.2.5	Signifikante Attribute	112

Zusammenfassung

In dieser Diplomarbeit wird das System PROXIMA vorgestellt, das in der Lage ist, ausgehend von Modelldaten graphische Abstraktionen zu erzeugen. Eine graphische Abstraktion ist die vereinfachte Darstellung eines in der Realität komplexeren Gegenstandes. Werden Details gezielt verringert, um z.B. eine Fokussierung auf einen bestimmten Bildbereich zu erreichen, so kann die Information vom Betrachter ungestörter aufgenommen werden. Auf diese Weise gelingt es, relevante von irrelevanten Einzelheiten zu trennen. Ausgehend von detaillierten 3D-Modelldaten erzeugt PROXIMA mithilfe von *Abstraktionstypen* vereinfachte Modelle. Die wesentlichen Abstraktionstypen sind das *Dimensionieren*, das *Verschmelzen* und das *Weglassen* von Teilobjekten. Bildgenerierungsverfahren (z.B. ein Projektionsalgorithmus oder ein Renderer) generieren aus abstrahierten Modellen dann eine Illustration, die auf der Bildebene durch die Abstraktionstypen *Verändern der Strichstärken* und *Vereinheitlichen* weiter bearbeitet werden kann.

Eine hohe Flexibilität wird dadurch erreicht, daß PROXIMA die Abstraktionen nicht vorberechnet, sondern einen inkrementellen Ansatz benutzt. Zum einen lassen sich so von ein und demselben Modell je nach Zielsetzung unterschiedliche Ergebnisse erzeugen. Zum anderen kann der Abstraktionsgrad während des Generierungsprozesses beeinflusst werden.

PROXIMA unterstützt zwei Betriebsmodi. Im interaktiven Betriebsmodus kann der Benutzer über eine graphische Schnittstelle abstrahierte Illustrationen generieren. Dazu wählt er die zu abstrahierenden Details direkt mit der Maus an und beeinflusst über verschiedene Parameter das Abstraktionsresultat. Im zweiten Betriebsmodus erzeugt PROXIMA Abstraktionen vollautomatisch. Die gewünschten Ergebnisse werden durch die Formulierung von Darstellungsbedingungen spezifiziert, die zu erfüllen sind. Mithilfe der Information über signifikante Attribute des Modells, propagiert PROXIMA dann für jedes Objekt des Modells eine Sichtbarkeitsbedingung. Auf dieser Basis wird anschließend der Abstraktionsgrad des Modells schrittweise erhöht.

PROXIMA ist das zur Zeit umfangreichste und flexibelste Werkzeug zur Erzeugung graphischer Abstraktionen. Trotz des experimentellen Charakters bieten sich aufgrund seiner Konzeption viele Anwendungsgebiete an. Im interaktiven Modus kann PROXIMA einen Graphiker oder technischen Zeichner beim Generieren schematischer Darstellungen unterstützen. Der vollautomatische Modus ermöglicht den Einsatz in der multimodalen Informationspräsentation.

Kapitel 1

Einleitung

Es ist nicht einfach, sich der Bedeutung des Begriffs der Abstraktion zu nähern, da seine vielfältige Verwendung die Eingrenzung erschwert. Der große Brockhaus [10] notiert unter Abstraktion schlicht:

„Philos.: allgemeiner Begriff; Sprachw.: Hauptwort, das etwas Nichtgegenständliches bezeichnet.“

Schlägt man das Verb *abstrahieren* nach, so ist zu lesen:

„Das Allgemeine vom Einzelnen absondern, zum Begriff erheben, verallgemeinern.“

Diese Fähigkeit der Begriffsbildung kennzeichnet eine bedeutende Leistung des menschlichen Geistes und ermöglicht es ihm, außergewöhnliche kognitive Leistungen zu erbringen. Der mächtige Prozeß der Kategorisierung und der Strukturierung von Wissen, zu dem der Mensch Abstraktionsmechanismen verwendet, spiegelt sich in vielen seiner Wirkungsbereiche wider.

So werden die meisten technischen Arbeitsprozesse durch Abstraktion hierarchisch organisiert, um effizienter abzulaufen. Beim Entwurf eines hochintegrierten Schaltkreises¹ wird z.B. die Funktion der Schaltung zunächst auf einer hohen Abstraktionsebene beschrieben. Schrittweise wird diese Beschreibung in niedrigere Abstraktionsebenen überführt, bis die unterste Ebe-

¹VLSI-Entwurf, siehe dazu [36].

ne der physikalischen Realisierung erreicht ist. Ein weiteres Beispiel im Bereich der Informatik sind objektorientierte Programmiersprachen (z.B. CLOS oder C++). Sie versetzen den Programmierer, gerade aufgrund ihrer Abstraktionsmechanismen, in die Lage, Aufgaben einfach darzustellen und zu lösen.

Auch in der Künstlichen Intelligenz werden Abstraktionsmechanismen untersucht, um sie zur Lösung von komplexen Problemen zu verwenden. Bei der Repräsentation von Wissen auf Computern wird, basierend auf kognitiven Modellen, Abstraktion als Strukturierungswerkzeug verwendet². Planungsansätze, die mit wiederverwendbaren Plänen aus einer Planbibliothek arbeiten³, können mithilfe eines Abstraktionsprozesses generellere Anfragen an die Planungsbibliothek formulieren. So werden auch Pläne gefunden, die erst nach einer geringfügigen Modifikation eingesetzt werden können. Abstraktionsmechanismen finden auch Verwendung bei der rechnergestützten, intelligenten Zusammenfassung von Texten⁴.

Neben diesen technisch orientierten Gebieten hat die Abstraktion, vor allem in der Kunst, durch Maler und Bildhauer⁵, seit Beginn dieses Jahrhunderts einen festen Platz eingenommen. Abstrakte Künstler versuchen ihre Empfindungen und Eindrücke durch abstrakte Konzepte in ihren Bildern und Skulpturen zu vermitteln. Die Wirkung ihrer Werke ist dabei abhängig vom Betrachter. Dieser hat oft gar nicht die Möglichkeit, den Abstraktionsprozeß konkret nachzuvollziehen und greift auf viele seiner eigenen Erfahrungen und Vorstellungen zurück. Einen Einblick in den künstlerischen Abstraktionsprozeß ermöglicht der spanische Maler Joan Miró. Sein im Jahre 1933 entstandenes Werk⁶ *Gemälde nach einer Collage* (Abbildung 1.1) abstrahiert von einer konkreten Vorlage, auf der technische Gegenstände abgebildet sind (Abbildung 1.2). Die so entstandene Abstraktion ist nicht funktional, aber dennoch nicht völlig losgelöst von der ursprünglichen Vorlage. Obschon Miró im gemalten Bild sehr stark abstrahiert, kann der Betrachter mithilfe der Collage jeden Gegenstand identifizieren, da die Lage der Objekte zueinander und ihre wesentliche Form erhalten bleibt. Außerdem werden gleichartige Objekte, wie z.B. die beiden Gefäßreihen, im Gemälde ähnlich dargestellt.

²Z.B. bei terminologischen Logiken [28].

³Siehe dazu [35].

⁴Einige Arbeiten dazu werden in [19] vorgestellt.

⁵Einen Überblick bietet [15].

⁶Die Abbildungen stammen aus [41].

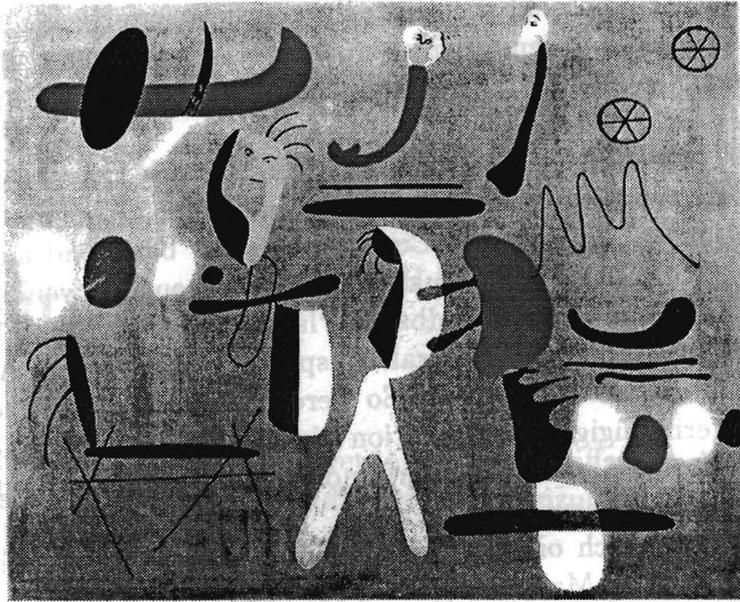


Abbildung 1.1: Joan Miró: *Gemälde nach einer Collage* (1933)

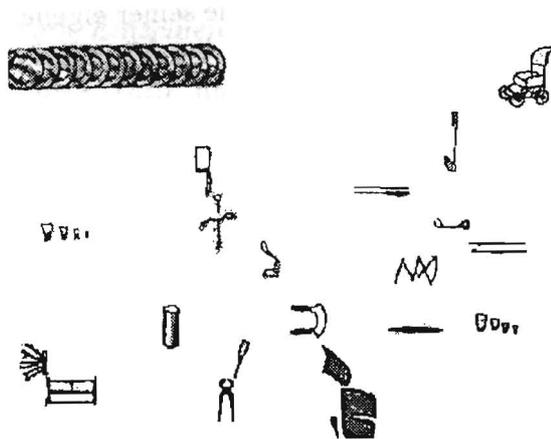


Abbildung 1.2: Vorbereitende Collage für das Gemälde in Abb. 1.1

Abstraktionen finden allerdings nicht nur in der Kunst Verwendung, sondern werden ebenso in Gebrauchsgraphiken eingesetzt. Der Graphikdesigner kann mithilfe der Abstraktion seine Arbeitsergebnisse übersichtlicher gestalten. Dabei möchte er aber Zweideutigkeiten vermeiden. Ihm liegt vielmehr daran, durch verschiedene graphische Mittel seine Illustration zu abstrahieren, ohne relevante Details zu unterschlagen. Dieser Umstand wird in Abbildung 1.3 und 1.4 verdeutlicht. Die erste Abbildung zeigt einen mikroskopisch vergrößerten und dann photographierten Teil eines Libellenflügels. Diese Form der Darstellung ist aber für viele Zwecke nicht geeignet. Der Betrachter wird durch viele Einzelheiten verwirrt und abgelenkt. Die abstrahierte Abbildung 1.4 ist wesentlich besser geeignet, um bestimmte Aspekte, z.B. die tragenden Teile des Flügels zu erläutern.

Unter Berücksichtigung der Aufgaben, die Illustrationen erfüllen sollen (siehe dazu auch Kapitel 2), kann durch den Einsatz von Abstraktionen ein übersichtlicheres Ergebnis erzielt werden.

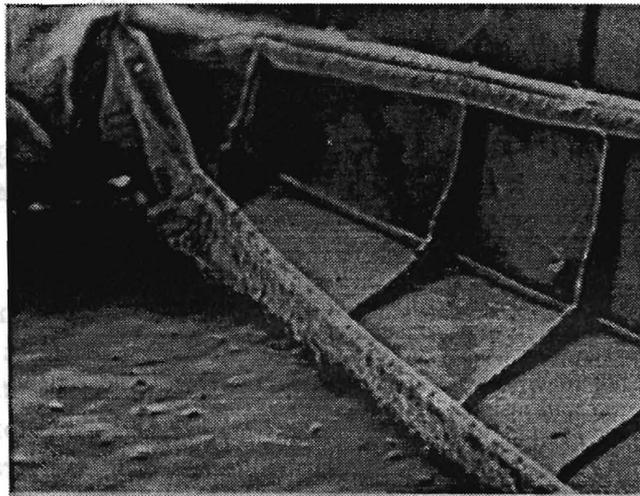


Abbildung 1.3: Die vergrößerte Darstellung eines Libellenflügelstücks.

⁶Die Abbildungen stammen aus [67].

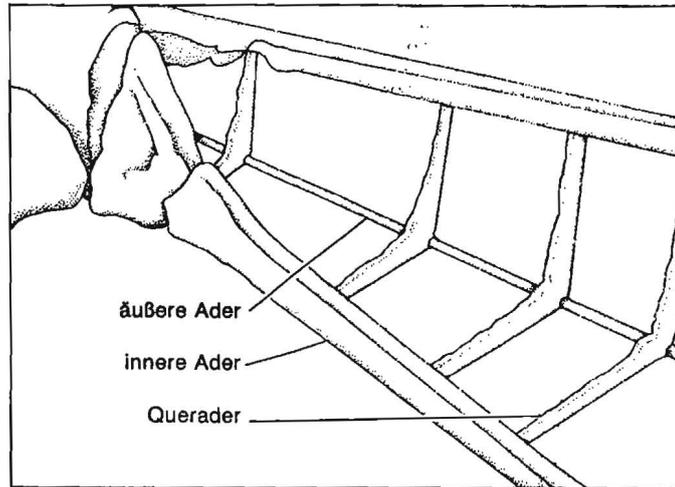


Abbildung 1.4: Die abstrahierte Darstellung des Libellenflügelstücks.

1.1 Ziel und Gegenstand der Arbeit

Die vorliegende Arbeit ist im Schnittbereich von Künstlicher Intelligenz und Graphikdesign angesiedelt. Sie untersucht, inwiefern graphische Abstraktionen automatisiert werden können, und welches Wissen für diesen Prozeß benötigt wird. Als Ausgangslage dienen 3D-Modelldaten, die z.B. aus einem CAD-System stammen könnten. Es sollen also Abstraktionsverfahren gefunden werden, die direkt mit diesen Modelldaten arbeiten. Bedingt durch die wenigen Vorarbeiten auf diesem Gebiet ist eine grundlegende Untersuchung der Abstraktionsziele und Abstraktionsmethoden notwendig. Im praktischen Teil der Arbeit wird darauf aufbauend das experimentelle Abstraktionssystem PROXIMA vorgestellt, das in der Lage ist, einfache Beispielabstraktionen zu erzeugen.

1.2 Aufbau der Arbeit

Das folgende Kapitel beschäftigt sich mit den verschiedenen Erscheinungsformen von graphischen Abstraktionen. Anhand von Bildbeispielen wird eine

Klassifikation von Illustrationen durchgeführt. In Kapitel 3 werden interdisziplinäre Grundlagen und die wenigen Vorarbeiten vorgestellt und diskutiert. In Kapitel 4 werden grundlegende Aspekte der Automatisierung der Abstraktion diskutiert, um dann aufbauend auf diesen Erkenntnissen, in Kapitel 5 das Abstraktionssystem PROXIMA vorzustellen. Kapitel 6 illustriert an verschiedenen Anwendungsbeispielen die Einsatzmöglichkeiten und Abstraktionsfähigkeiten des Systems. Kapitel 7 wendet sich an den Anwendungsprogrammierer und den Schluß bildet Kapitel 8 mit einer Zusammenfassung und einem Ausblick auf Erweiterungsmöglichkeiten des Systems.

Kapitel 2

Graphische Abstraktionen

In diesem Kapitel werden unter der Berücksichtigung diverser Fallbeispiele hauptsächlich drei Fragen beantwortet:

- Welche Ziele kann man mithilfe graphischer Abstraktionen erreichen?
- Wie wird eine graphische Abstraktion realisiert?
- Welche Auswirkungen haben Randbedingungen auf die Abstraktionen?

Nach der Vereinbarung und Erläuterung diverser Begriffe wird zunächst im Abschnitt *Abstraktionsziele* die erste Frage behandelt. Daraufhin werden die Mittel (die sogenannten *Abstraktionstypen*) untersucht, durch die Abstraktionen entstehen (siehe zweite Frage). Ein weiterer Teil des Kapitels beschäftigt sich mit dem Kontext, in dem Abstraktionen verwendet werden. Er definiert *Randbedingungen*, die Einfluß auf die Auswahl der Abstraktionstypen haben. Zum Abschluß wird der Zusammenhang zwischen Abstraktionszielen, Abstraktionstypen und Randbedingungen erläutert.

2.1 Begriffserklärung

Es folgen einige kurze Begriffsvereinbarungen, um Mißverständnisse von vornherein zu vermeiden.

- *Objekte* sind graphische Objekte, d.h. entweder CAD-Objekte oder einfach nur Bildobjekte.

- *(3D-)Modelle* sind CAD-Modelle.
- *Modellrepräsentation* ist die Art, in der 3D-Modelle rechnerintern repräsentiert werden. Geläufige Repräsentationen sind das Drahtrahmenmodell, das Flächenmodell (b-reps) oder die Konstruktion mit Raumprimitiven (CSG). Einen Überblick verschiedener Repräsentationsformen geben [24] und [22].
- Die *Zusammenbauhierarchie* beschreibt den hierarchischen Aufbau eines Modells.
- Die *materiellen Attribute* eines Objekts bezeichnen seine Form, seine Farbe, sein Material und seine Größe.
- *Bildschirmkoordinaten* bezeichnen die Koordinaten eines Punktes nach seiner Projektion in die Bildebene.
- *Weltkoordinaten* sind die dreidimensionalen Koordinaten eines Punktes im Raum.
- Die *Darstellung* eines Objekts, ist immer die Ansicht eines Objekts auf einem zweidimensionalen Medium (Papier oder Bildschirm).
- Der *Bildentstehungsprozess* bezeichnet den Generierungsprozess einer Darstellung, ausgehend von 3D-Modelldaten, und beinhaltet sowohl die beliebige Manipulation der Modelle, als auch den nötigen Projektionsvorgang.
- *Abstraktion* wird im folgenden immer als graphische Abstraktion aufgefaßt.
- *Abstraktionsalgorithmen* sind Verfahren, die durch Modifikation von 3D-Modellen und ihren Projektionen graphische Abstraktionen erzeugen.
- *Abstraktionsdomänen* sind 3D-Modelldaten, die Ausgangsbasis für Abstraktionsalgorithmen sind.
- Das *Abstraktionsergebnis* ist das Resultat eines Abstraktionsalgorithmus.

2.2 Abstraktionsziele

Bei der Untersuchung welche Zielsetzung mit Abstraktionen verfolgt werden, wird deutlich, daß dies vornehmlich *semantische* Ziele sind. Diese Zielsetzung nimmt Bezug auf den Bildinhalt, und man unterscheidet im wesentlichen zwei semantische Abstraktionsziele:

1. *Reduzierung des Bildinhaltes*
mittels Beschränkung auf relevante Details. So wird der Betrachter nicht durch übermäßige Information belastet. Es kann auch eine Fokussierung auf wesentliche Bildinhalte erreicht werden. Die Abbildungen 1.3 und 1.4 in Kapitel 1 sind ein gutes Beispiel für eine Fokussierung durch Abstraktion.
2. *Erzeugung eines prototypischen Vertreters einer Kategorie*
durch Mischung von Eigenschaften einer ganzen Klasse in einem einzelnen Objekt. Diese Form der Abstraktion entspringt einem Modell aus der Begriffspsychologie (siehe Kapitel 3.1.1), in dem eine Kategorie durch einen sogenannten Prototypen charakterisiert wird. In technischen Dokumentationen können mithilfe von Prototypen Handlungen visualisiert werden. Ein typisches Beispiel ist in Abbildung 2.1 zu sehen. Ein Schraubenzieher wird verwendet, um das Drehen einer Schraube zu verdeutlichen. Wäre dieser Schraubenzieher sehr speziell so bestünde die Gefahr, daß der Betrachter zu der Überzeugung käme, er müsse genau dieses spezielle Werkzeug verwenden. Dieser Eindruck kann vermieden werden, wenn die Darstellung des Werkzeugs so gewählt wird, daß es eine ganze Klasse von Schraubenziehern vertritt.

Außer den *semantischen Abstraktionszielen* existiert noch eine weitere Gruppe von Abstraktionszielen, die nicht ausschließlich mit der Generierung von Graphiken zusammenhängt. Abstraktion verringert die Komplexität der Graphiken und damit auch die Komplexität der ihnen zu Grunde liegenden Datenstrukturen. Somit kann der Wunsch nach Abstraktion in Hinblick auf die Ressourcen der verwendeten Computertechnologie auch *technische* Gründe haben.

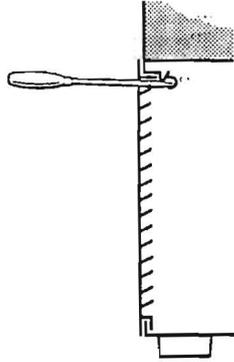


Abbildung 2.1: Der Schraubenzieher als prototypischer Vertreter.¹

1. *Auskommen mit beschränkten Ressourcen*
wird durch Abstraktion ermöglicht, wenn Restriktionen berücksichtigt werden müssen. Dies kann die Beschränkung der Darstellung auf bestimmte Farben, Strichstärken oder Auflösung sein (dies wird oft durch das Ausgabemedium vorgegeben, wie z.B. Drucker oder Bildschirm).
2. *Effizienzsteigerung*
durch Vereinfachung der Modelldaten mithilfe der Abstraktion. So ergeben sich Effizienzsteigerungen bei der photorealistischen Darstellung von Modellen (z.B. Rendern oder Radiosity), da weniger Modellteile in diesem Prozeß berücksichtigt werden müssen. In Kapitel 6 wird ein Vorschlag gemacht, wie Animationsberechnungen durch Vereinfachung der Modelle beschleunigt werden können.
3. *Strukturvergleiche*
anhand der abstrahierten geometrischen Modelldaten. Ist die Abstraktion eindeutig, so können abstrahierte Modelle für schnelle Vergleiche anstatt der ursprünglichen Modelle verwendet werden (siehe dazu auch Abschnitt 3.2.1). Desweiteren lassen sich mithilfe von Strukturvergleichen Bilddatenbankzugriffe beschleunigen. Diese Idee wird in Kapitel 6 vorgestellt.

¹Die Abbildung stammt aus [16].

2.3 Abstraktionstypen

In diesem Abschnitt wird untersucht, wie Abstraktionsziele graphisch realisiert werden. Ausgangspunkt der Untersuchung ist wiederum Bildmaterial, das aus technischen Dokumentationen stammt. Folgende elementare *Abstraktionstypen* konnten identifiziert werden.

1. Weglassen von Objekten

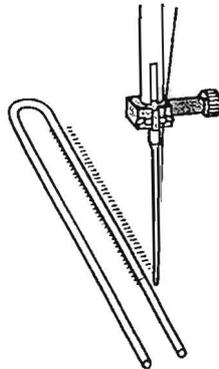


Abbildung 2.2: Unwichtige Teile der Nähmaschine wurden *weggelassen*.²

Unter dem *Weglassen* versteht man das Eliminieren eines Objekts. Abbildung 2.2 wurde der Bedienungsanleitung einer Nähmaschine entnommen. Es wird ein mögliches Stichmuster erklärt. Um diesen Sachverhalt zu unterstreichen, wurden nur relevante Teile der Nähmaschine abgebildet und unwichtige Objekte weggelassen. Diese starke Form der Abstraktion findet nur in einem Kontext, der Ambiguitäten ausschließt, eine sinnvolle Anwendung.

Es können aber auch komplexe Objekte durch *Weglassen* kleiner Einzelteile schrittweise vereinfacht werden.

²Die Abbildung stammt aus [46].

2. Verschmelzen von Objekten

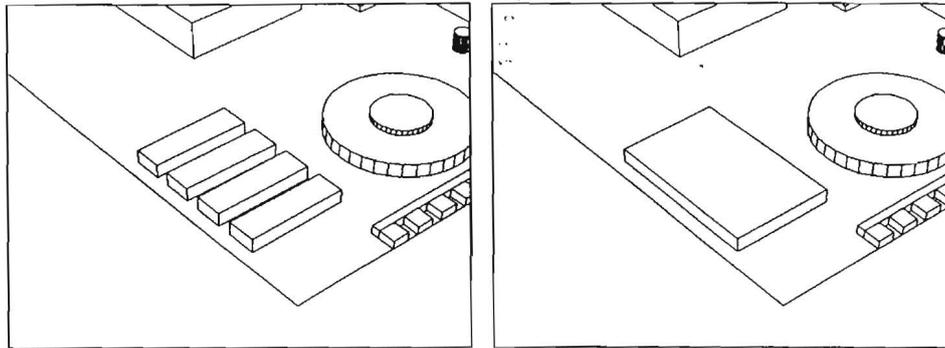


Abb.2.3 (a): Die IC-Bank auf einer Platine...

Abb.2.3 (b): ... wird durch *Verschmelzen* abstrahiert.

Das *Verschmelzen* ist ein Abstraktionstyp, der aus räumlich getrennten Objekten ein einziges Objekt erzeugt. Dieser Typ gestattet es, Attribute der ursprünglichen Objekte im neuen Objekt zusammenzufassen. Betrachten wir hierzu Abbildung 2.3(a). Sie zeigt den Ausschnitt einer Platine, auf der sich verschiedene Bauteile befinden, unter anderem eine Reihe von ICs. In Abbildung 2.3(b) ist diese IC-Bank durch den Abstraktionstyp *Verschmelzen* abstrahiert worden.

3. Verändern der Dimensionen von Objekten

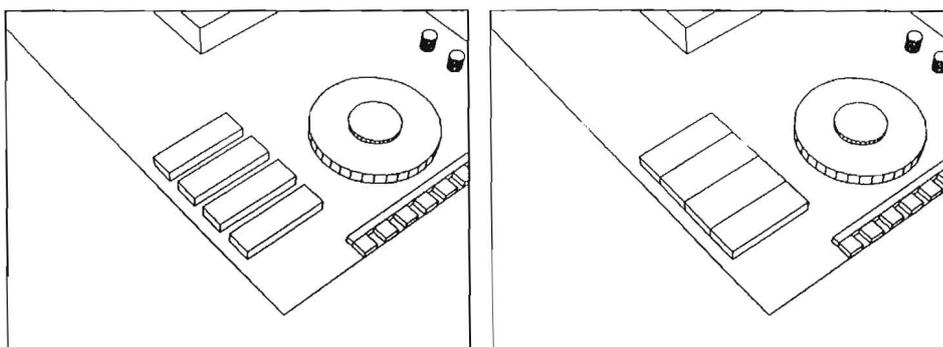


Abb.2.4 (a): Die IC-Bank auf einer Platine...

Abb.2.4 (b): ... wird durch *Wachsen* abstrahiert.

Durch *Verändern der Dimensionen* können Objekte gruppiert werden, ohne daß sie verschmolzen werden. Dabei „wachsen“ benachbarte Flächen der Objekte aufeinander zu, bis sie sich berühren. Nach dem *Gestaltgesetz der Nähe* (wird in Abschnitt 3.1.2 behandelt) werden dicht beieinander liegende Objekte eher als Gruppe gesehen als Objekte die weiter voneinander entfernt liegen. Dieser Abstraktionstyp abstrahiert ähnlich dem *Verschmelzen*, ist aber in seiner Wirkung auf das Resultat nicht ganz so stark. Die Bildsequenz 2.4 verdeutlicht dies am Beispiel der IC-Bank noch einmal.

4. Vereinheitlichen von Objekten

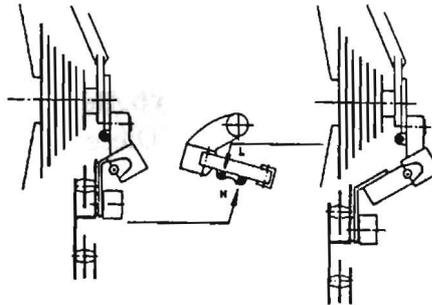


Abbildung 2.5: Strichstärken werden vereinheitlicht.³

Werden mehrere Objekte mit derselben Strichstärke dargestellt, so spricht man von *Vereinheitlichen*. Die Justierung einer Fahrradkettenschaltung wird in Abbildung 2.5 illustriert. In der gesamten Graphik wird nur eine Strichstärke verwendet, um die Bildobjekte zu vereinheitlichen.

5. Skalieren von Objekten

Unter dem *Skalieren* ist ein Abstraktionstyp zu verstehen, der die relative Größe von Objekten beeinflusst. So können Hintergrundobjekte verkleinert und Objekte im Vordergrund vergrößert werden. In Abbildung 2.6 werden die Fassung des Zigarettenanzünders und der entsprechende

³Die Abbildung stammt aus [59].

Stecker im Verhältnis zum Armaturenbrett vergrößert dargestellt, da sie bezüglich der Bildaussage besonders wichtig sind.

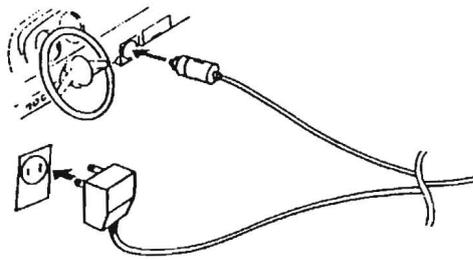


Abbildung 2.6: Bildinhalte werden skaliert.⁴

6. Verändern der Strichstärken

kann dazu verwendet werden, Bildinhalte hervorzuheben bzw. in den Hintergrund zu legen. Objekte, die in den Vordergrund gerückt werden sollen, werden in dickeren Strichen gezeichnet als Hintergrundobjekte.

7. Symbolisieren von Objekten

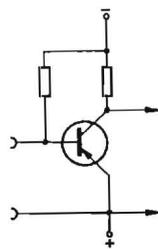


Abbildung 2.7: Die symbolische Darstellung eines Schaltkreises.⁵

⁴Die Abbildung stammt aus [56].

⁵Die Abbildung stammt aus [6].

Beim *Symbolisieren* wird sowohl von geometrischen, als auch von stofflichen Objekteigenschaften abstrahiert. Dies entspricht einem Ersetzungsvorgang. Bildobjekte werden durch symbolische Darstellungen ersetzt. Das Symbolisieren wird oft in stark technisch orientierten Dokumentationen verwendet. Ein typisches Beispiel ist die Darstellung eines Schaltkreises, wie er in Abbildung 2.7 zu sehen ist.

2.4 Einfluß von Randbedingungen auf die Wahl von Abstraktionstypen

Die Wahl der Abstraktionstypen wird in erster Linie durch das gewünschte Abstraktionsziel bestimmt. Allerdings gibt es eine Reihe von Randbedingungen, die, unabhängig von den gewünschten Abstraktionszielen, Einfluß auf die Auswahl der Abstraktionstypen haben. Unterschiedliche Randbedingungen können die Aussagekraft einer Abstraktion erheblich verändern und sollten deshalb berücksichtigt werden. In dieser Arbeit werden drei unterschiedliche Arten von Randbedingungen vorgestellt.

- **Konsistenzbedingungen**

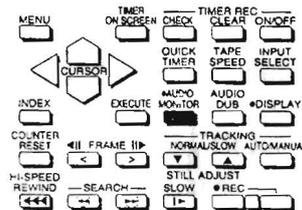


Abbildung 2.8: Konsistente Abstraktion innerhalb eines Bildes.⁶

Die *Konsistenzbedingungen* bewirken, daß gleichartige Objekte nicht grundlos verschieden abstrahiert werden. Abbildung 2.8 verdeutlicht dies. Die Aufmerksamkeit des Betrachters wird auf den invertierten

⁶Die Abbildung stammt aus [47].

Knopf gelenkt. Die übrigen Knöpfe, die den Hintergrund bilden, werden alle im selben Detaillierungsgrad dargestellt.

- **Persistenzbedingungen**

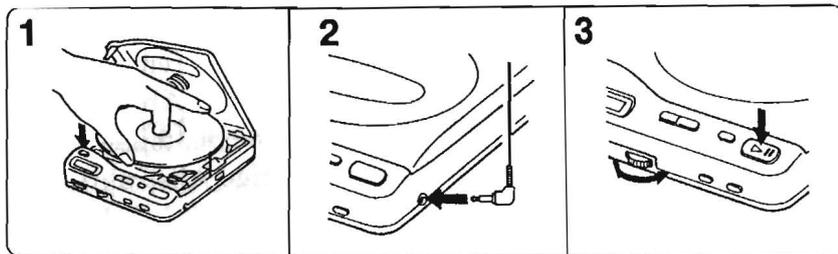


Abbildung 2.9: Persistenzbedingungen müssen beachtet werden.⁷

Die *Persistenzbedingungen* verhindern, daß ein Bildinhalt, dem am Ende einer Bildfolge Bedeutung zukommt, nicht zu Beginn der Folge beliebig starker Abstraktion unterzogen wird. Dies würde gegen „das Prinzip der kontinuierlichen Veränderung“ verstoßen. Durch die Berücksichtigung von Persistenzbedingungen kann ein solcher Bruch vermieden werden. Abbildung 2.9 erläutert einige Schritte der Bedienung eines CD-Spielers. Dabei werden nacheinander verschiedene Knöpfe betätigt. Die Knöpfe dürfen zwar abstrahiert werden, aber das Weglassen eines dieser Knöpfe wäre eine Verletzung der Persistenz.

- **Bedingungen durch den situativen Kontext**

Unter dem *situativen Kontext* versteht man äußere Faktoren, wie z.B. den bildlichen oder den textuellen Kontext. Auch das Hintergrundwissen des Betrachters fließt mit ein. So ist es möglich, dem Betrachter eine Illustration mit starken Abstraktionen zu präsentieren, wenn in einem Textteil zusätzlich Information dargeboten wird. Auch sind zum Beispiel abstrahierte Darstellungen für Experten verschieden von solchen, die für Laien bestimmt sind.

⁷Die Abbildung stammt aus [56].

2.5 Zusammenfassung

In den vorangegangenen Abschnitten dieses Kapitels wurden Ziele vorgestellt, die durch graphische Abstraktion erreicht werden können. Es wurde erklärt, durch welche graphischen Mittel abstrahierte Illustrationen entstehen und welche Rolle Randbedingungen spielen. In der nachfolgenden Tabelle ist zu sehen, wie die Wahl und Kombination der Abstraktionstypen mit den verfolgten Abstraktionszielen in Verbindung steht. Am oberen Rand der Tabelle sind die Abstraktionstypen, am rechten Rand sind die Abstraktionsziele aufgelistet. Ein Kreuz in der Tabelle bedeutet, daß der entsprechende Abstraktionstyp verwendet werden kann, um das zugehörige Abstraktionsziel zu verwirklichen. Beim Betrachten der Tabelle stellt man fest, daß die Abstraktionstypen *Weglassen* und *Verschmelzen* am universellsten einsetzbar sind. Im Gegensatz dazu findet z.B. das *Symbolisieren* eher seltene Verwendung.

Typen Ziele	Weg- lassen	Ver- schmel- zen	Dimen- sionie- ren	Verein- heit- lichen	Skalie- ren	Ver- ändern	Symbo- lisieren
Reduzierung des Bildinhaltes	X	X	X	X	X	X	X
Fokussieren (Vordergrund)					X (größer)	X	X
Fokussieren (Hintergrund)	X	X	X	X	X (kleiner)		X
Prototypen- generierung		X	X		X		
Ressourcen- beschränkung				X		X	
Effizienz- steigerung	X	X					
Struktur- vergleiche	X	X					

Kapitel 3

Relevante Vorarbeiten

Die graphische Abstraktion wird in der Literatur zum technischen Zeichnen und zur Künstlichen Intelligenz eher spärlich thematisiert. Aus diesem Grund soll dieses Kapitel nicht nur eine wissenschaftliche Basis für die vorliegende Arbeit sein, sondern auch eine Sammlung von Ideen und Ansätzen zu diesem Thema. So entsteht ein Kompromiß insofern, als sowohl die wenigen, eng verwandten Systeme, als auch die wichtigsten interdisziplinären Grundlagen präsentiert und besprochen werden.

Der erste Abschnitt widmet sich der Begriffsforschung und den Gestaltungsrichtlinien, die im Graphik-Design Berücksichtigung finden. Im zweiten Abschnitt werden Vorarbeiten aus dem Bereich der Informationsverarbeitung vorgestellt. Besondere Aufmerksamkeit gilt dabei dem System APEX [20], welches als einziges System bisher in der Lage war, umfassendere graphische Abstraktionen zu erzeugen.

Damit sich diese Besprechung der Grundlagen übersichtlich gestaltet, erfolgt am Ende jedes Teilabschnittes eine kurze Diskussion.

3.1 Relevante Arbeiten aus Psychologie und Graphik-Design

3.1.1 Begriffspsychologie

Die Begriffspsychologie beschäftigt sich mit den kognitiven Vorgängen der Begriffsbildung und Kategorisierung. Genau diese Vorgänge gilt es zu beachten,

wenn man graphische Abstraktionen von Objekten vornehmen will. Der Versuch einer Abstraktion kann nämlich als gescheitert angesehen werden, wenn der Betrachter sich keinen Begriff von ihr machen kann. Es muß also darauf geachtet werden, daß, trotz einer durch die Abstraktion bedingte Veränderung der ursprünglichen Objekte, die Begriffsbildung nicht behindert wird. Es gelingt sogar durch geeignete Abstraktionen, den begriffsbildenden Prozeß zu beschleunigen. Ein weiterer Grund, die Erkenntnisse, die in diesem Teilgebiet der Psychologie gewonnen wurden, zu studieren, resultiert aus der Tatsache, daß die empirischen Untersuchungen zum größten Teil anhand von Bildmaterial durchgeführt wurden.

Es folgt zunächst eine Zusammenstellung der Forschungsergebnisse, die der überarbeiteten Habilitationsschrift von Thomas Eckes [17] entnommen wurden. Anschließend wird erläutert, inwiefern diese Ergebnisse in der graphischen Abstraktion berücksichtigt werden müssen.

3.1.1.1 Die klassische Theorie

Die Grundidee der klassischen Theorie läßt sich bis auf den aristotelischen Substanzbegriff zurückführen. Mit Substanz ist hier der Wesensgehalt gemeint, durch den die Klassenzugehörigkeit eines Objekts eindeutig festgelegt wird. Der aristotelischen Logik zufolge gilt das Bivalenzprinzip (Zweifertigkeitsprinzip), d.h. jede Aussage der Form:

Objekt A gehört der Klasse C an

ist entweder wahr oder falsch. Daraus folgt, daß es eine Reihe von Merkmalen gibt, durch die ein Objekt eindeutig kategorisiert werden kann. Ferner ist jedes Objekt einer Kategorie gleichsam gut geeignet, diese zu vertreten, denn jedes Mitglied einer Kategorie besitzt genau diese charakteristischen Merkmale.

Eine Kategorisierung kann also, nach der klassischen Theorie, anhand eines Merkmalvektors vorgenommen werden. Demzufolge dürfen bei einer graphischen Abstraktion nur solche Merkmale verändert oder weggelassen werden, die nicht Elemente des Merkmalvektors sind, um die Kategorisierung nicht zu gefährden.

Die experimentelle psychologische Forschung zum Thema Begriffslernen wurde in den 20er und 30er Jahren von Ach [1], Fisher [23] und Hull [31] begonnen. Die *klassische Begriffstheorie* fand dann auch in verschiedenen anderen wissenschaftlichen Disziplinen Verwendung und galt bis zur Hälfte des 20. Jahrhunderts als unangefochten bester Ansatz zur Begriffsforschung.

Als einer der Ersten kritisierte Wittgenstein [66] die klassische Begriffstheorie. Er widersprach der Forderung der klassischen Theorie nach einer Menge von begriffsdefinierenden Merkmalen, die alle notwendig und gemeinsam hinreichend eindeutig über die Begriffszugehörigkeit eines Objekts entscheiden. Als Beispiel führt Wittgenstein den Begriff *Spiel* an. Er argumentiert, daß es keine Menge von Merkmalen gibt, die allen Elementen dieses Begriffs gemeinsam ist. Hieraus folgt, daß zwei Elemente einer Kategorie nicht ein gemeinsames Merkmal besitzen müssen. Sehr wohl besteht aber eine Verkettung von Merkmalen quer durch eine Kategorie. Dieses Konzept nennt Wittgenstein die *Familienähnlichkeit* innerhalb einer Kategorie. Das folgende Beispiel soll diesen Begriff verdeutlichen: Eine Kategorie hat fünf Objekte o_1, o_2, o_3, o_4 und o_5 , wobei jedes dieser Objekte durch den Besitz oder Nicht-Besitz von fünf Merkmalen X_1, X_2, X_3, X_4 und X_5 beschrieben sei. Die Matrix X (Objekt-Prädikat-Tafel) habe folgendes Aussehen:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
o_1	0	1	1	1	1	0	0	0
o_2	1	0	1	1	1	0	0	0
o_3	1	1	0	1	1	0	0	0
o_4	1	1	1	0	1	0	0	0
o_5	1	1	1	1	0	0	0	0
\vdots								
o_9	1	0	0	0	0	1	1	1

Jedem Objekt o_i fehlt das Merkmal X_i , so daß keines der fünf Merkmale allen Objekten gemeinsam ist. Allerdings besteht zwischen jeweils zwei Objekten ein gewisses Maß an Übereinstimmung. Objekt o_9 besitzt die Merkmale X_1, X_6, X_7 und X_8 und hat damit *kein* Merkmal mit Objekt o_1 gemeinsam.

Trotzdem sind sie durch eine *Familienähnlichkeit* miteinander verbunden.

Eine weitere Kritik kam vom Linguisten Labov [39], der zeigte, daß Kategorien unscharf sein können. In einer Untersuchung präsentierte Labov einer Reihe von Versuchspersonen (Vpn) die Umrißzeichnungen verschiedener Gefäße und bat um eine Benennung. Abbildung 3.1 zeigt einige dieser Gefäße.

Das mittlere Gefäß in der oberen Reihe wurde von allen Vpn als *Tasse*

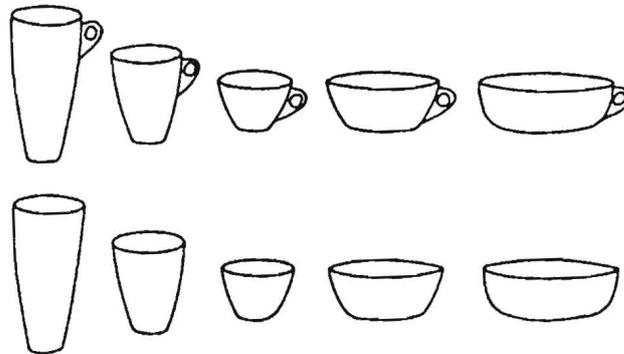


Abbildung 3.1: Beispiel zur Vagheit linguistischer Kategorien.¹

benannt. Durch Veränderung des Höhe/Breite Verhältnisses wurden zunehmend häufiger die Begriffe *Schüssel* bzw. *Vase* genannt. Fehlte den Gefäßen der Henkel (untere Reihe), dann sank die Häufigkeit für die Benennung *Tasse* stärker bei Veränderungen der Breite und Höhe. Labov zeigte ferner, daß auch funktionale Merkmale Einfluß auf die Wortwahl hatten. So bat er die Vpn, sich vorzustellen, daß jemand das Gefäß in der einen Hand halte und mit der anderen Zucker darin umrühre, um dann daraus Kaffee zu trinken. Daraufhin wurde die Bezeichnung *Tasse* häufiger als im neutralen Kontext gewählt. Die perzeptuellen Merkmale wurden je nach funktionalem Kontext verschieden gewichtet, und die einzelnen Elemente somit von Fall zu Fall unterschiedlich kategorisiert.

Heute ist unbestritten, daß eine reine syntaktische Definition nicht genügt, um alltagssprachliche Kategorisierungsprozesse zu erklären.

¹Die Abbildung stammt aus [39].

3.1.1.2 Die Gradiententheorie

Als Alternative zur klassischen Begriffstheorie bildete sich eine Theorie, die wesentliche Kritikpunkte aufnahm und vor allem auch pragmatische Aspekte der Kategorisierung enthielt.

Die zentralen Annahmen der Gradiententheorie sind:

1. Eine Liste definierender Merkmale reicht i.a. nicht aus, um Objekte eines Begriffes korrekt zu beschreiben. Es wird vielmehr davon ausgegangen, daß innerhalb einer Kategorie bestimmte Merkmale eine relativ höhere Auftretungswahrscheinlichkeit besitzen als andere oder mithilfe derer eine Abgrenzung zu anderen Klassen erleichtert wird. Es wird jedoch nicht ausgeschlossen, daß es eine Liste von Merkmalen gibt, die alle Objekte einer Kategorie gemeinsam haben. Diese Merkmale werden *charakteristische Merkmale* genannt.
2. Charakteristische Merkmale haben bei der Kategorisierung von Objekten unterschiedliches Gewicht. Dieses Gewichte ist abhängig von Kontexteinflüssen und somit nicht statisch.
3. Objekte sind unterschiedlich tauglich, um als Vertreter ihrer Klasse zu agieren. Das Ausmaß, in dem ein Objekt als typisch für seine Kategorie gilt wird als *Typikalität* bezeichnet. Das Kontinuum, auf dem sich Objekte als mehr oder weniger typisch anordnen lassen, heißt *Typikalitätsgrad*.

Die dritte Annahme gibt der Gradiententheorie ihren Namen.

Innerhalb der Gradiententheorie gibt es mehrere Varianten und Modelle, die aus zwei grundlegenden Fragen resultieren:

1. Wird ein Prozeß der Abstraktion von Information über Exemplare eines Begriffes angenommen, der zu einer summarischen Repräsentation führt, oder wird darauf verzichtet?
2. Werden die charakteristischen Merkmale von Begriffsexemplaren als kategorial (diskret) oder als dimensional (kontinuierlich) konzipiert?

Die Beantwortung der ersten Frage ergibt zwei unterschiedliche Modellvarianten. Die sogenannte *Prototypentheorie* [43] bejaht diese Frage und geht

davon aus, daß charakteristische Merkmale einer Kategorie abstrahiert, und dann als ein kognitiver Prototyp im Gedächtnis gespeichert werden.

Im Gegensatz dazu steht die *Exemplartheorie* [11], welche einen solchen Prozeß völlig verneint und davon ausgeht, daß alle bekannten Exemplare einer Kategorie im Gedächtnis gespeichert werden.

Die zweite Frage befaßt sich mit den charakteristischen Merkmalen der Begriffsexemplare. Diskrete Merkmale sind rein binär. Das heißt, eine bestimmte Eigenschaft kann einem Objekt zugeordnet werden oder nicht. Ist ein Merkmal hingegen dimensional, so kann es innerhalb eines bestimmten Bereiches beliebige Werte annehmen. Wichtig ist diese Unterscheidung vor allem im Hinblick auf die Präzisierung des Prototypenkonzepts. Der sogenannte *Modal-Prototyp* ist eine Liste von gewichteten Merkmalen, die am häufigsten innerhalb einer Kategorie auftreten. Liegt allerdings ein dimensionaler Ansatz zugrunde, so erhält man den *Durchschnittsprototypen*. Schon 1883 beschäftigte sich Galton [25] mit einem ähnlichem Prototypenkonzept. Er versuchte, durch Übereinanderlegen von Portraitphotographien charakteristische Merkmale eines Personenkreises zu ergründen. In diesem Fall wollte er die spezifischen Gesichtszüge von Straftätern untersuchen. Dazu erzeugte er ein Durchschnittsportrait, mithilfe dessen neue Gesichter klassifiziert werden sollten².

Die Ähnlichkeit eines Objekts zu einem Prototypen, bzw. in der Exemplartheorie zu allen gespeicherten Exemplaren, entscheidet über die Typikalität des Objekts.

3.1.1.3 Diskussion

Durch die graphische Abstraktion werden Objekte vereinfacht, indem einzelne Attribute verändert bzw. ganz entfernt werden. Entscheidend ist dabei, daß Zielvorgaben trotz oder gerade durch die Abstraktion verwirklicht werden können. Die Fähigkeit des Betrachters, auch abstrahierte Objekte korrekt klassifizieren zu können, spielt dabei eine entscheidende Rolle. Die Begriffspsychologie bietet im Grunde zwei verschiedene Theorien an (die *klassische Theorie* und die *Gradiententheorie*), um dies zu gewährleisten. Dabei muß abgewogen werden, welche Theorie am besten geeignet ist, rechnergestützte Abstraktionen zu ermöglichen.

²Dabei spielten verschiedene Dimensionen eine Rolle, wie Länge der Nase, Größe des Mundes o. ä.

Die *klassische Theorie* bietet sich aufgrund der einfachen Umsetzung an: Eine wohldefinierte Menge von diskriminierenden Attributen bestimmt die Zugehörigkeit eines Objekts zu einer Klasse. Dies läßt sich relativ einfach für ein Abstraktionssystem formalisieren.

Wie aber gezeigt wurde, existiert berechtigte Kritik an dieser Theorie. Pragmatische Aspekte der Objekte müssen berücksichtigt werden.

Beim Versuch, Abstraktionen auf Basis der *Gradiententheorie* zu erzeugen, darf die Exemplartheorie ausgeklammert werden, da es nicht realistisch erscheint, alle Objekte einer Kategorie zu speichern.

Dies gelingt am ehesten durch die Prototypentheorie, indem jeweils ein spezielles Objekt eine ganze Kategorie vertritt. Soll ein Objekt noch richtig kategorisiert werden, so muß sich die Abstraktion des Objekts an dem Prototypen der Kategorie orientieren und darf sich in ihrer Darstellung nicht zu weit von ihm entfernen.

3.1.2 Design und Gestalttheorie

Der Graphik-Designer muß bei der Ausübung seiner Tätigkeit eine Zielgruppe berücksichtigen, für die seine Arbeit bestimmt ist. Er muß eine Nachricht visuell so umsetzen, daß diese von den potentiellen Betrachtern seiner Graphiken richtig interpretiert wird.

Im Prinzip ist mit jeder Anfertigung einer Graphik, die Gegenstände aus der realen Welt darstellt, ein Abstraktionsvorgang verbunden. Daß Gegenstände nicht so wiedergegeben werden können, wie sie sich dem Menschen in der realen Welt präsentieren, ist leicht einzusehen. Dies hängt mit dem zweidimensionalen Medium (Papier und Bildschirm) sowie den Mitteln (Pinsel, Farbe usw.) zusammen, die dem Designer im Normalfall zur Verfügung stehen. Ein Blick in ein Graphik-Design Standardwerk (z.B. [42]) macht deutlich, daß der Graphik-Designer darüber hinaus oft gezwungen ist, von den realen Gegenständen noch stärker zu abstrahieren, um mit wenigen Mitteln das Wesentliche zu zeigen. Dabei spielt die Kontur eines Gegenstandes eine entscheidende Rolle. Diese imaginäre Grenzlinie zwischen Flächen, Körpern und Hintergrund genügt oft, um einen Gegenstand zu erkennen³.

³Vorausgesetzt, ein ausreichend starker Kontrast ermöglicht eine deutliche Unterscheidung vom Hintergrund.



Sowohl das Kartenzeichen, das Gesicht und das Telefon werden als Figur wahrgenommen, obwohl nur die Umrißlinien gezeichnet wurden. Unter Berücksichtigung der Zielsetzung, daß seine Bilder korrekt interpretierbar bleiben, kann der Graphik-Designer also eine Abstraktion realisieren, indem er nur die Kontur eines Gegenstandes darstellt.

Es gibt aber auch noch andere Aspekte, die im Designprozeß von Graphiken berücksichtigt werden müssen. Gerade bei der graphischen Abstraktion, die besonders die Vorstellungskraft des Betrachters beansprucht, darf die grundsätzliche Wirkung von Bildern auf den Betrachter nicht vernachlässigt werden. Die Kognitionspsychologie befaßt sich mit diesem Problem. Basierend auf physiologischen Fakten und empirischen Ergebnissen über die menschliche visuelle Wahrnehmung, werden dort verschiedene psychologische Aspekte der Wirkung optischer Reize eingehend untersucht. Dabei sind, unter dem Gesichtspunkt der graphischen Abstraktion, besonders die Teilgebiete der Ästhetikforschung und der Gestaltpsychologie interessant.

Die Ästhetikforschung beschäftigt sich mit der „Schönheit“ von Bildern. In [7] wird festgestellt, daß ein quantitatives Problem der Ästhetik existiert, das eng mit der Komplexität der Bilder verbunden ist. Zuwenig Komplexität wirkt aufgrund der geringen Stimulanz langweilig auf den Betrachter. Zuviel Komplexität bewirkt Unbehagen, das auf eine Überreizung zurückzuführen ist. Komplexitätsfaktoren können sein:

- Anzahl der Elemente
- Verschiedenartigkeit der Elemente

- Anordnung der Elemente, d.h. der Grad dessen, wie verschiedene Elemente als Einheit aufgefaßt werden. .

Aber nicht nur die Komplexität eines Bildes beeinflußt die Wahrnehmung. Auch die sogenannte Bildstimmung wirkt sich auf die Bildwahrnehmung aus. Die Faktoren der Bildstimmung sind:

- Bildaufbau (Anordnung der Elemente)
- Bildstil (längerfristige konstante Merkmale)
- Farbe

Die Gestaltpsychologie erforscht die Prägnanz von Bildern. Die Untersuchung von Gestaltgesetzen und Gestaltfaktoren ([3] und [33]) geben Aufschluß darüber, wie ein Betrachter höherwertige Gruppierungen von Sachverhalten wahrnimmt. Im folgenden sind die wichtigsten bisher gefundenen Gestaltgesetze zusammengefaßt:

- Gesetz der Nähe: Benachbarte Objekte werden zusammengeschlossen wahrgenommen.
- Gesetz der Ähnlichkeit: Es besteht die Tendenz, nicht benachbarte, in Gestalt und Form ähnliche Objekte als zusammengehörend anzusehen.
- Gesetz des glatten Verlaufs: Untereinander kreuzende Linien werden so wahrgenommen, daß der Verlauf jeder einzelnen Linie möglichst wenige und kaum abrupte Knicke enthält.
- Gesetz der Geschlossenheit: Teilweise verdeckte Bildelemente werden im verdeckten Bereich durch den Betrachter logisch nachvollziehbar fortgesetzt.

3.1.2.1 Diskussion

Die Erkenntnisse dieses Abschnitts ermutigen ausdrücklich, auch aus ästhetischen Gründen graphische Abstraktionen vorzunehmen, falls sich zu viele Elemente in einem Bild befinden. Eine Möglichkeit besteht darin, Darstellungen von Gegenständen auf ihre Konturen zu reduzieren. In Abschnitt 2.3

wurden die Abstraktionstypen *Verschmelzen* und *Weglassen* vorgestellt, mit deren Hilfe eine solche Abstraktion realisiert werden kann.

Aus der Gestaltpsychologie kann man Bedingungen für Abstraktionstechniken ableiten. Sie sollten die Gestaltgesetze berücksichtigen. Vor allem das Gestaltgesetz der Nähe und das der Ähnlichkeit spielen eine wichtige Rolle. Es scheint sinnvoll, die durch diese Gesetze beschriebenen Gruppen während der Abstraktion beizubehalten. Mithilfe der Abstraktionstypen *Verändern der Dimensionen* und *Verschmelzen* kann diese Tatsache berücksichtigt werden, indem nah beieinanderliegende Objekte zu größeren Einheiten zusammengefaßt werden.

3.2 Relevante Arbeiten aus dem Bereich der Informatik

Dieser Abschnitt stellt zunächst das einzige System vor, das in der Lage ist, graphische Abstraktionen zu erzeugen und widmet sich dann zwei weiteren verwandten Systemen.

3.2.1 Abstraktionen im System APEX

Einen ersten Ansatz zur automatischen Erzeugung von abstrahierten Illustrationen, in denen 3D-Objekte unterschiedlicher Abstraktionen dargestellt werden, präsentierte Feiner 1985 im Rahmen seiner Doktorarbeit [20]. Er bezeichnet seine Abstraktionen als sogenannte *approximations*, also als Annäherungen der ursprünglichen Objekte.

Die Beschreibung der Objektmodelle erfolgt, wie in CAD Systemen üblich, durch eine hierarchische Beschreibung. Feiner benutzt dabei in seiner Modellwelt nur achsenparallele Quader, die in ihren Dimensionen variabel sind. Aus diesen Grundobjekten werden komplexere Objekte durch eine Zusammenbauhierarchie definiert.

Die grundlegende Idee ist es, nun eine Abstraktionshierarchie (ähnlich der Zusammenbauhierarchie) zu entwerfen, in der jeder Knoten eine Abstraktion seiner Söhne enthält. In Abbildungen 3.2 ist auf der linken Seite die Zusammenbauhierarchie einer Knopfleiste zu sehen. Rechts erkennt man eine

mögliche, dazugehörige Abstraktionshierarchie. Die Blätter der Zusammenbauhierarchie enthalten vier Knöpfe und die darunter liegende Leiste. Die Knöpfe 1 bis 3 werden in der ersten Abstraktionsebene (repräsentiert durch die tiefste Ebene von Knoten in der Abstraktionshierarchie) zu einer Leiste abstrahiert, während der vierte Knopf noch unverändert bleibt. Erst in der obersten Abstraktionsebene wird auch Knopf 4 in die Abstraktion miteinbezogen, und das Resultat ist ein einzelner länglicher Quader.

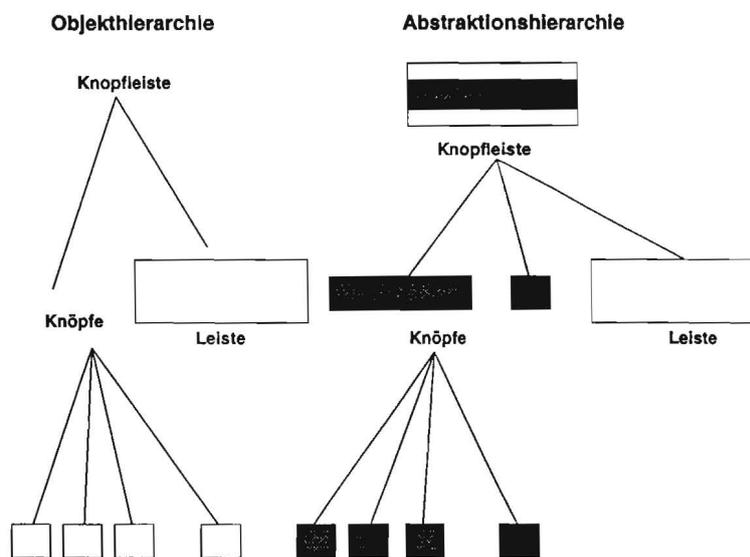


Abbildung 3.2: Die graphische Abstraktion einer Knopfleiste.⁴

Feiner stellt fünf Forderungen auf, die ein geeignetes Abstraktionsverfahren erfüllen soll:

- *Handhabung verschiedener Materialarten*
In APEX können Objekte aus verschiedenen Materialien bestehen. Dies muß das Abstraktionsverfahren berücksichtigen, denn zwei Objekte mit völlig unterschiedlichen stofflichen Eigenschaften sollten nicht zu einem Objekt abstrahiert werden.

⁴Abbildung stammt sinngemäß aus [20].

- *Benutzen der bestehenden Zusammenbauhierarchie*
Die bestehende Objekthierarchie sollte von einem Abstraktionsalgorithmus berücksichtigt werden, denn sie liefert Information über Gruppierungen der Objekte und ihre relative Lage im Raum.
- *Flexible Reaktion auf Veränderungen der Zusammenbauhierarchie*
Dies ist nötig, da in APEX die Zusammenbauhierarchie nicht starr ist, sondern Veränderungen unterworfen ist. Wird ein Modell modifiziert, z.B. durch die Bewegung eines Schalters, so sollte das Abstraktionsverfahren Kohärenzkriterien ausnutzen, um aufwendige Rechnungen einzusparen.
- *Berücksichtigung des Zusammenspiels von Teilen*
Einige Teile sind so entworfen, daß sie zusammenpassen. Ihre Abstraktionen sollen sinnvollerweise ebenfalls zusammenpassen.
- *Unabhängigkeit von der Wahl der Perspektive*
Dies ermöglicht es, abstrahierte Darstellungen eines abstrahierten Modells aus verschiedenen Perspektiven zu erhalten, ohne dabei jedesmal eine neue Abstraktion zu berechnen.
- *Abgeschlossenheit der Abstraktion*
Die Abstraktionen sollten aus den gleichen Primitiven bestehen wie die Originalobjekte. Bei Feiner sind dies ausschließlich achsenparallele Quader.
- *Eindeutigkeit der Repräsentation*
Zwei Objekte mit der gleichen Zusammenbauhierarchie sollen auch in einer gleichen Abstraktion resultieren.

Wie schon erwähnt, kann die Zusammenbauhierarchie durch einen Baum (im folgenden: *Objektbaum*) charakterisiert werden. Das Abstraktionsverfahren erzeugt nun eine ähnliche Datenstruktur, den sogenannten *Abstraktionsbaum*. Dieser Abstraktionsbaum wird aus dem Objektbaum berechnet, und beide Bäumen werden an ihren Knoten und Blättern miteinander verbunden. So wird jedem Teilobjekt (repräsentiert durch einen Knoten im Objektbaum) eine Abstraktion zugeordnet. Der Abstraktionsbaum wird von den Blättern aufwärts aufgebaut. Die Blätter des Abstraktionsbaums sind identisch mit denen des Objektbaums. Die inneren Knoten des Abstraktionsbaums jedoch,

repräsentieren Abstraktionen der inneren Knoten des Objektbaums. In Abbildung 3.3 ist ein Objektbaum mit seinem dazugehörigen Abstraktionsbaum zu sehen. Quadratische Knoten und durchgezogene Kanten repräsentieren den Objektbaum. Die runden Knoten und die gestrichelten Linien repräsentieren den Abstraktionsbaum. Die Blätter beider Strukturen sind identisch und enthalten die unveränderten Objekte.

Jeder Knoten des Abstraktionsbaums ist eine Abstraktion von einem oder mehreren Söhnen. Die Zusammenhänge zwischen Abstraktionen und Originalobjekten werden durch Bögen verdeutlicht.

Die inneren Knoten des Objektbaums repräsentieren zwar stoffliche Eigenschaften, diese sind aber nicht explizit aufgeführt, sondern werden von unten nach oben innerhalb des Baumes vererbt. Im Gegensatz dazu sind in allen Knoten des Abstraktionsbaums stoffliche Eigenschaften explizit repräsentiert. Jeder dieser Knoten repräsentiert eine Abstraktion, die in einer adäquaten Modellrepräsentation zur Verfügung steht (z. B. als Drahtrahmenmodell in einem CAD System). Das von Feiner vorgestellte Abstraktionsverfahren besteht im wesentlichen aus drei Schritten, um mehrere Knoten zu einer neuen Abstraktion zusammenzufassen:

1. Objekte (hier: achsenparallele Quader), die nahe nebeneinander liegen, verändern ihre Größe derart, daß die Lücke zwischen beiden geschlossen wird. Die zugewandten Flächen der Quader „wachsen“ sozusagen aufeinander zu. Dies entspricht dem Abstraktionstyp *Verändern der Dimensionen* (siehe Abschnitt 2.3).
2. Besitzen diese zwei Quader ähnliche stoffliche Eigenschaften, so entsteht ein neues Objekt mit den Dimensionen der beiden alten. Diese Abstraktionstechnik wurde in Abschnitt 2.3 als *Verschmelzen von Objekten* vorgestellt.
3. Zuletzt werden noch eventuelle Objekte eliminiert, wenn sie kleine oder sehr flache Teile eines größeren Objekts sind.

Die Durchführung des ersten Schritts bestimmt zunächst für eine Fläche, ob sich eine andere Fläche in ihrer Nähe befindet. Der Suchraum beschränkt sich dabei auf bestimmte Abstände in X-, Y- und Z-Richtung. Eine andere Fläche befindet sich dann in ihrer *Nähe*, wenn sie den durch diese Werte bestimmten Quader schneidet. Die Fläche mit dem geringsten Abstand ist die

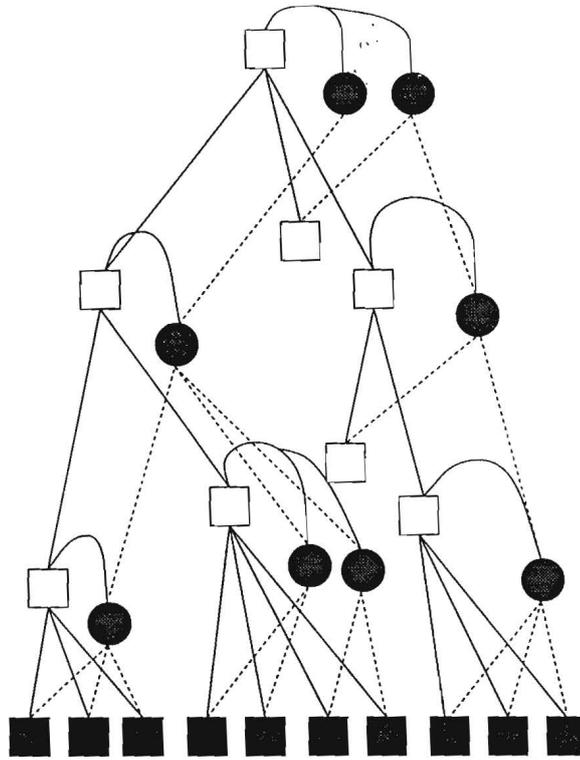


Abbildung 3.3: Ein Hierarchiebaum mit dem dazugehörigen Abstraktionsbaum.⁵

sogenannte *Nachbarsfläche*. Diese Nachbarsfläche wird nun für alle Flächen des zu abstrahierenden Objekts berechnet, und in einem nächsten Schritt „wachsen“⁶ diese Nachbarsflächen paarweise aufeinander zu, so daß sich die zugehörigen Quader berühren. Dieser Prozeß wird in Abbildung 3.4 verdeutlicht. Alle Lücken zwischen benachbarten Objekten schließen sich. Im zweiten Schritt werden Objekte dann miteinander *verschmolzen*, wenn sie annähernd die gleichen stofflichen Eigenschaften⁷ besitzen, und wenn die sich berühren-

⁵Die Abbildung stammt sinngemäß aus [20].

⁶Der Betrag, um den die einzelne Fläche sich bewegt, ist abhängig von ihrer relativen Größe.

⁷Diese Eigenschaften werden unter Berücksichtigung eines simplen Beleuchtungsmo-

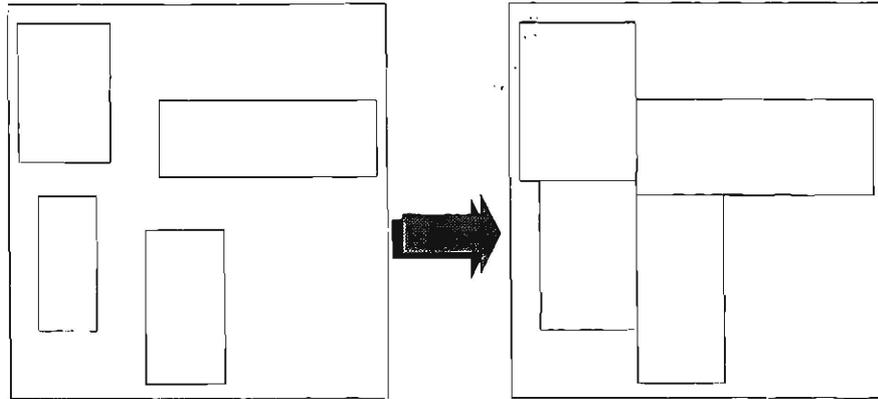


Abbildung 3.4: Die Lücken zwischen den Quadern werden durch „Wachsen“ geschlossen.

den Flächen nicht von zu unterschiedlicher Größe sind. Das neue Objekt besitzt dann die Dimension der Vereinigung der beiden alten Objekte und eine gemittelte Stoffeigenschaft.

In Abbildung 3.5 werden mehrere aneinandergrenzende Quader verschmolzen. Dabei ist der Größenunterschied der nebeneinander liegenden Flächen von wesentlicher Bedeutung. Wird ein bestimmter Wert überschritten, so werden die jeweiligen Objekte nicht miteinander verschmolzen. Dieser Umstand erzeugt die Abstufung in zwei Blöcke.

Zwei Arten von Objekten werden schließlich im dritten Schritt eliminiert: Erstens kleine Objekte, die neben einem größeren Objekt liegen (Abbildung 3.6) und nur eine Nachbarsfläche besitzen. Zweitens größere, aber sehr flache Objekte, die ähnliche stoffliche Eigenschaften besitzen wie ein größeres, danebenliegendes Objekt (Abbildung 3.7). Abstrahierte Darstellungen des Objekts werden nun generiert, indem nicht alle Blätter des Abstraktionsbaums dargestellt werden, sondern schon in geringerer Tiefe die abstrahierten Objekte in den Bildentstehungsprozeß miteinfließen.

Variierende, abstrahierte Darstellungen von demselben Objekt lassen sich realisieren, indem in verschiedenen Bereichen des Abstraktionsbaums in unterschiedlichen Tiefen Knoten ausgewählt werden. Dabei gilt: je niedriger die

dells (siehe [24]) numerisch berechnet und dann verglichen.

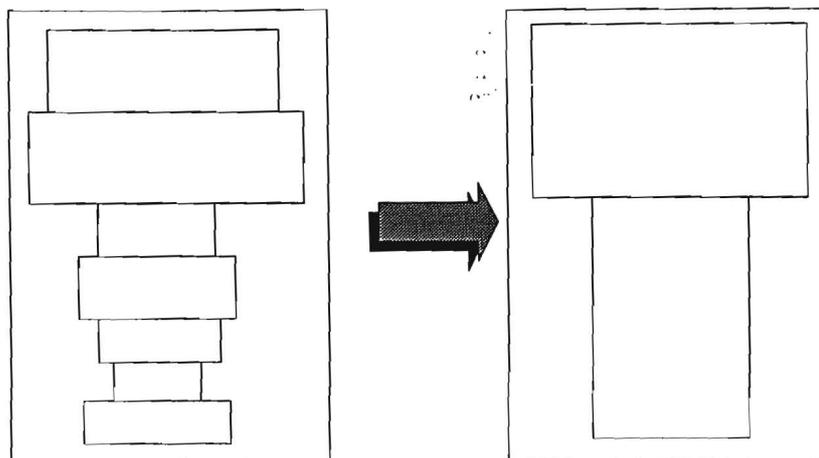


Abbildung 3.5: Eine Quaderreihe wird zu zwei neuen Objekten verschmolzen.

Tiefe eines Knotens, desto geringer sein Abstraktionsgrad, bzw. desto höher seine Detailtreue bzgl. des Originalobjekts. Die so entstandenen, abstrahierten Darstellungen können dann in der Informationspräsentation [21] (siehe auch Abschnitt 6.3.1) verwendet werden. Soll z.B. ein bestimmtes Teilobjekt augenfällig plaziert werden, so wird es detailgetreuer als der Hintergrund dargestellt. Ein spezieller Teil des Algorithmus vergleicht dazu die Abstraktionen von Vorder- und Hintergrundobjekten zunächst auf hohem Abstraktionsniveau (d.h. die jeweiligen Knoten liegen sehr weit oben im Abstraktionsbaum). Hier mögen die Unterschiede noch nicht so groß sein, wenn man bedenkt, daß in einer Welt von achsenparallelen Quadern und mit den beschriebenen Abstraktionsmethoden die stärkste Abstraktion eines komplexen Objekts nur ein Quader sein kann. Das Verfahren wählt nun für die Darstellung der Vordergrundobjekte einen tiefer gelegenen Knoten im Abstraktionsbaum aus, bei dem die Unterschiede deutlicher zu erkennen sind⁸.

Dieses Vergleichsverfahren kann auch dazu verwendet werden, beim Erzeugen von photorealistischen Darstellungen in speziellen Fällen Rechenzeiten einzusparen. Die Objekthierarchie (und damit auch der zugehörige Ab-

⁸Dies wird über eine numerische Funktion entschieden, die die Dimensionen der Objekte und ihre stofflichen Eigenschaften berücksichtigt.

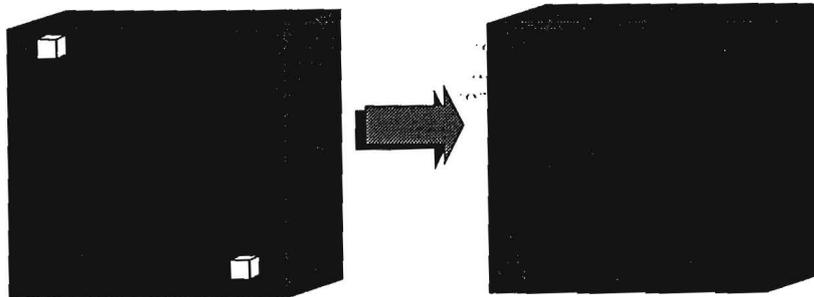


Abbildung 3.6: Relativ kleine Objekte werden eliminiert.

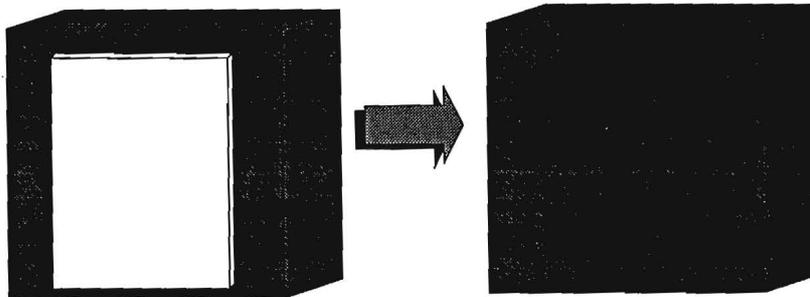


Abbildung 3.7: Flache und dünne angrenzende Objekte werden entfernt.

straktionsbaum) ändert sich, wenn z.B. ein Teil durch ein anderes ersetzt wurde. Durch einen Vergleich des so neu entstandenen Abstraktionsbaums mit seinem alten Vorgängern kann ermittelt werden, welche Bildbereiche neu berechnet werden müssen und welche übernommen werden können⁹.

3.2.1.1 Diskussion

Feiners Arbeit ist als Pionierleistung in der rechnergestützten Generierung von graphischen Abstraktionen anzusehen. Er hat einen Abstraktionsansatz gefunden, indem er Objekte durch *Wachsen* und *Verschmelzen* zu abstrak-

⁹Dies ist ein allgemeines Verfahren zum Speichern von Massendaten, wie z.B. ein inkrementelles Backup oder Verfahren zur Bildkompression (siehe dazu auch [38]).

teren Objekten gruppiert bzw. zusammenfaßt. Für die Bedürfnisse seiner Anwendungen genügt eine vorausberechnete Abstraktionshierarchie, in der er verschiedene Abstraktionsebenen zur Auswahl hat.

Als problematisch erweist sich dieser Ansatz, wenn abstrahierte Darstellungen unter erweiterten Gesichtspunkten gewünscht werden. Man verdeutliche sich dies anhand des schon eingeführten Beispiels mit der Knopfleiste (Abbildung 3.2). In der Abstraktionshierarchie existieren drei verschiedene Abstraktionen der Knopfleiste, die ausgewählt werden können. Will man jedoch die Aufmerksamkeit des Betrachters auf Knopf 1 lenken, so braucht man eine Abstraktion, die Knopf 1 unverändert läßt, in der aber die anderen Knöpfe abstrahiert werden (siehe Abbildung 3.8).

Auch beim *Weglassen* von Objekten kann es zu ungewünschten Ergebnissen

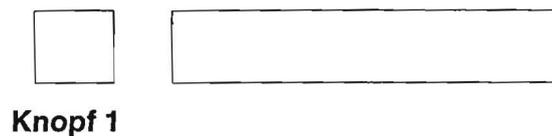


Abbildung 3.8: Durch Abstraktion der Knopfleiste wird Knopf 1 hervorgehoben.

kommen. Kleine Details sind nicht selten unabhkömmlich, um eine korrekte Klassifizierung vorzunehmen und dürfen daher nicht als irrelevant eingestuft werden. Auch flache dünne Flächen können charakteristisch für ein Objekt sein und es kann, wie die Abbildungen 3.9 und 3.10 zeigen, durch das Entfernen solcher Flächen wichtige Information verloren gehen. Dem Betrachter wird der Blick auf den dunklen Würfel durch eine dünne weiße Fläche versperrt. Daher bietet sich diese Szene dem Betrachter wie in 3.10(a) dar. Wird nun diese dünne Fläche entfernt, so entsteht für den Betrachter der verwirrende Eindruck aus 3.10(b).

Diese Punkte können von Feiners Abstraktionsansatz deshalb nicht berücksichtigt werden, da der Abstraktionsvorgang nur aufgrund syntaktischer Merkmale (Distanz der Objekte voneinander, Dimensionen, sowie materielle Eigenschaften) arbeitet. Hier ist ein Verfahren vorteilhaft, das auch pragmatische Gesichtspunkte berücksichtigt und die Abstraktionen erst dynamisch bei Bedarf erzeugt, um möglichst präzise den zusätzlichen Anforderungen zu genügen.

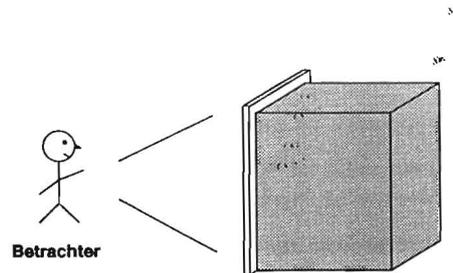


Abbildung 3.9: Der dunkle Würfel wird durch die dünne Fläche verdeckt.

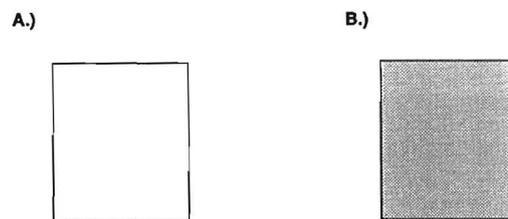


Abbildung 3.10: Das Entfernen der dünnen Fläche kann zu verwirrenden Ergebnissen führen.

3.2.2 Ein System zur Generierung von Skizzen

In [18] wird ein Skizzen-Renderer beschrieben, der von CAD-Modellen ausgehend in der Lage ist, skizzenhafte Zeichnungen zu erzeugen. Anders als bei Render-Systemen herkömmlicher Funktionsweise, die darauf abzielen möglichst präzise physikalische Bedingungen innerhalb einer Bildszene nachzustellen, wird bei diesem Ansatz versucht, ein Bild des Modells zu erzeugen, das dem einer menschlichen Zeichnung nahekommt. Die Argumentation für diesen Ansatz basiert auf zwei Punkten:

- Kommunikative Ziele sollen besser vermittelt werden.
- Einfachere Strukturen ergeben einen Effizienzgewinn, der es ermöglicht, den interaktiven Umgang mit Bildmaterial zu verbessern.

Der Skizzen-Renderer erzeugt mithilfe eines BSP¹⁰-Tree eine vektororientierte Linienbeschreibung, mit der die exakten Modelle ungenau dargestellt werden können. Dazu werden je nach gewünschtem Detaillierungsgrad mehr oder weniger Liniensegmente zur Darstellung gebracht (siehe dazu in Abbildung 3.11 den Deckenbereich). Das System verdeutlicht darüber hinaus Schattenwürfe und Hervorhebungen durch die Schraffierung von Flächen. Effektgeneratoren erzeugen zusätzlich Bewegungslinien bzw. künstliche Pflanzen und erlauben es, bestimmte Bildbereiche zu fokussieren. Die Skizzen

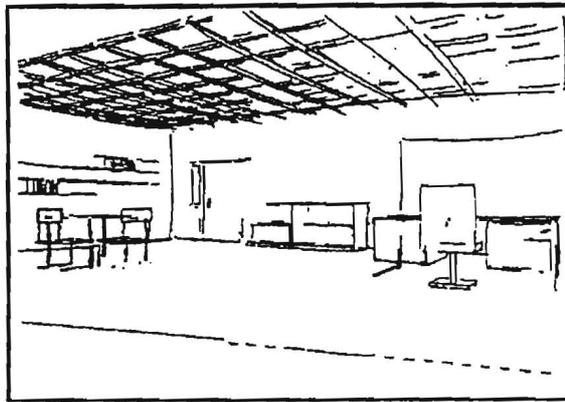


Abbildung 3.11: Eine mit dem Skizzen-Renderer realisierte Graphik.¹¹

entstehen im engen Dialog mit dem Benutzer: Er entscheidet, welche Bildbereiche abstrahiert werden und welche nicht. Durch die Verwendung einer *Sichtbarkeitsinformation* für jeden Bildbereich kann dabei interaktiv und direkt manipulativ gearbeitet werden.

Einsatzgebiete sind Architekturskizzen und Zeichnungen im medizinischen Bereich.

¹⁰*Binary Partition Tree*; dieses Verfahren wird z.B. auch in [45] verwendet.

¹¹Die Abbildung stammt aus [18].

3.2.2.1 Diskussion

Mit dem Skizzen-Renderer können aus exakten Modellen Skizzen generiert und somit abstrahierte Darstellungen erzeugt werden. Allerdings handelt es sich um ein Verfahren, dessen Parameter ausschließlich vom Benutzer gesetzt werden; eine Entscheidungshilfe wird nicht gewährt.

Darüber hinaus unterliegt der vektororientierte Ansatz einer Beschränkung: Eine Abstraktion, bei der Details der Modelle verändert werden, findet nicht statt. Dadurch können Attribute wie Farbe, Form und Struktur nicht im Abstraktionsprozeß berücksichtigt werden. In [20] wird z.B. gezeigt, daß auch herkömmlich gerenderte abstrahierte Modelle für wirkungsvolle Illustrationen verwendet werden können.

3.2.3 Ein System zur Konvertierung von CAD-Modellen in FEM-Modelle

Neue Bauteile werden zunehmend mithilfe von CAD-Systemen konstruiert und entworfen. Die Belastungen, denen das fertige Werkstück im Gebrauch ausgesetzt ist, können durch eine rechnergestützte FEM¹²-Analyse überprüft werden. So können Konstruktionsfehler erkannt und behoben werden. Die Modelle, mit denen FEM-Analysen durchgeführt werden, unterscheiden sich jedoch beträchtlich von den CAD-Modellen. Insbesondere sind FEM-Modelle weniger detailreich und häufig spielen Volumina keine Rolle. Die direkte Weiterverwendung der CAD-Daten ist deshalb nicht möglich und eine, normalerweise per Hand durchgeführte Konvertierung wird notwendig.

Das System TEA (Transformator EUCLID-ANSYS¹³ [8]) automatisiert diese Konvertierung der CAD-Modelle in FEM-Modelle. Ein Teil des System ersetzt Körper des CAD-Modells durch Flächen und *abstrahiert* so vom Volumen. Weitere Entwicklungsstufen des Systems sollen auch in der Lage sein, kleine Details zu entfernen und Symmetrieeigenschaften des CAD-Modells zur Abstraktion auszunutzen.

¹²Finite Elemente Methode

¹³EUCLID ist ein CAD- und ANSYS ein FEM-System.

3.2.3.1 Diskussion

Durch TEA werden CAD und FEM-Systeme effizient gekoppelt. Die Abstraktionsmechanismen, die zur Konvertierung verwendet werden (sollen) ähneln den Abstraktionstypen *Dimensionieren* und *Weglassen* der graphischen Abstraktion, die in Abschnitt 2.3 vorgestellt wurden. Trotzdem darf dies nicht darüber hinwegtäuschen, daß mit der graphischen Abstraktion zum Teil ganz andere *semantische* Ziele verfolgt werden (siehe Abschnitt 2.2). So spielen z.B. kleine Details eines Modells bei der FEM-Analyse keine entscheidende Rolle und können daher weggelassen werden. Bei graphischen Abstraktionen gilt dies nicht grundsätzlich, denn kleine Teile können entscheidend für die richtige Kategorisierung eines Objekts sein.

3.2.4 Ein System zur funktional basierten Objekterkennung

Die Güte einer Graphik ist offensichtlich abhängig von der Interpretation des potentiellen Betrachters. Deutliche und bekannte Bildobjekte unterstützen diesen Vorgang. Bei graphischen Abstraktionen stellt sich hier das Problem, daß die Abstraktion eines real existierenden Objekts in den meisten Fällen in einem fiktiven künstlichen Objekt resultiert. Daß Abstraktionen dennoch verwendet werden können, hängt mit der Fähigkeit des Menschen zusammen, Objekte einzuordnen, die er zuvor nie gesehen hat. Trotzdem bewegen sich Abstraktionen oft auf einem Grat zwischen besserem und schlechterem Verständnis.

Die Motivation sich in diesem Abschnitt mit der Objekterkennung zu beschäftigen, beruht auf zwei Feststellungen:

- Die Erzeugung von Abstraktionen, die gewissen Forderungen genügen (wie z.B. „Abstraktion ist erkennbar als Objekt X“), ist verwandt mit dem *umgekehrten* Prozeß der Bildererkennung („Erkenne ich dieses Objekt als Objekt X?“). Techniken aus der Bildererkennung können im Abstraktionsprozeß verwendet werden, um Forderungen im Abstraktionsergebnis zu berücksichtigen.
- Zusätzlich kann durch Objekterkennung eine Evaluierung der Abstraktionen durchgeführt werden. So ist es möglich den Abstraktionsvorgang als *generate and test* Verfahren zu realisieren.

Die *funktional basierte* Objekterkennung beruht auf dem Prinzip, ein Objekt nicht nur auf Grund syntaktischer Merkmale zu erkennen (wie z.B. Form, Volumen oder Farbe), sondern darüber hinaus zu versuchen, aus diesen Merkmalen Schlüsse über die Funktionalität des Objekts oder seiner Einzelteile zu ziehen. Mithilfe dieser Information wird dann das zu untersuchende Objekt klassifiziert.

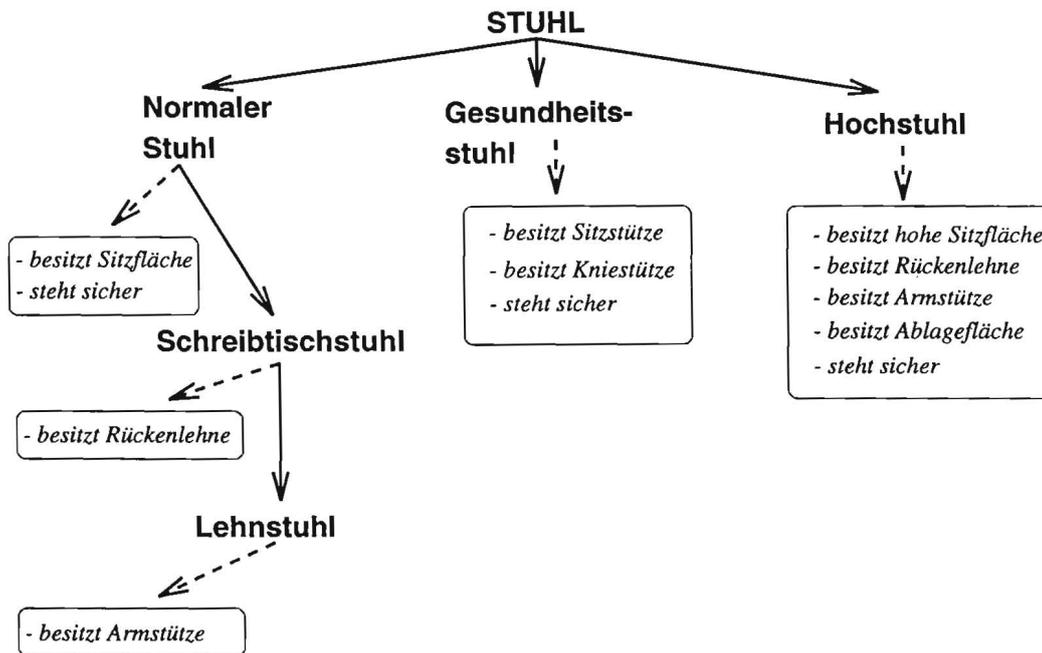


Abbildung 3.12: Die Begriffshierarchie und die Funktionalität der Objekte.

GRUFF2 [57] von Star und Bowyer ist ein Objekterkennungssystem für Möbelstücke. Das System erkennt Objekte aus vier verschiedenen Basiskategorien (Tisch, Stuhl, Bett und Regal). Die einzelnen Objekte sind als 3D-Modelle durch b-reps repräsentiert.

Interessant ist die Kombination aus prozeduralem und explizitem Wissen, die zur Lösung des Erkennungsproblems gewählt wurde. Grundsteine sind sogenannte Wissensprimitive. Dies sind Prozeduren, die bestimmte Eigenschaften des Modells untersuchen, wie Nähe der Objekte zueinander, Stabilität und Schwerpunkte oder Freiräume zwischen Objektteilen.

Das explizite Wissen über die Funktionalität der Objektkategorien wird durch ein Framesystem repräsentiert. Z.B. wird von einem einfachen Stuhl verlangt, daß er eine *Sitzfläche* zur Verfügung stellt und im *Gleichgewicht* steht. In Abbildung 3.12 wird gezeigt, wie sich die Kategoriehierarchie auf speziellere Mitglieder auswirkt. Es werden dann einfach neue Forderungen hinzugenommen (zum Beispiel muß ein Lehnstuhl zusätzlich eine *Stütze für den Rücken* besitzen). Überprüft werden diese Bedingungen an einem Objekt durch ein Wissensprimitiv oder durch die Kombination von mehreren Wissensprimitiven.

Zur Kontrolle der Ergebnisse wurden eine Reihe von empirischen Untersuchungen durchgeführt, die eine hohe Übereinstimmung zwischen menschlichen und maschinellen Einordnungen signalisieren.

3.2.4.1 Diskussion

Die Arbeit von Stark und Bowyers benutzt Wissensprimitive, um Funktionseigenschaften der untersuchten Objekte abzuleiten. Die verschiedenen Funktionseigenschaften eines Objekts werden in einer Begriffshierarchie gespeichert. Dieses Verfahren ist auch für die Abstraktion denkbar, wobei dann hierarchisch gespeichert werden müßte, welche Funktionseigenschaften für ein Objekt charakteristisch sind. Die Wissensprimitive würden während der Abstraktion dafür sorgen, daß wichtige Funktionseigenschaften der Objekte auch in der Abstraktion erhalten bleiben.

Ein funktional basiertes Objekterkennungssystem ist insbesondere geeignet, um Abstraktionsergebnisse zu überprüfen. Dabei werden allerdings nur Bilddaten verarbeitet. Zusätzlicher Kontext, z.B. ergänzender Text, fließt nicht in den Erkennungsprozeß mit ein.

3.2.5 Ein System zur „Verschönerung“ von Graphiken

Es existieren bei der automatischen Abstraktion und bei der rechnergestützten Verschönerung von mit der Hand gezeichneten Graphiken durch ein System ähnliche Probleme.

Eine automatische Verschönerung von Graphiken ist überall dort sinnvoll, wo computergestütztes präzises Zeichnen praktiziert wird. Es ist mit Graphikeditoren zwar prinzipiell möglich, genau zu zeichnen, allerdings ist dies meist

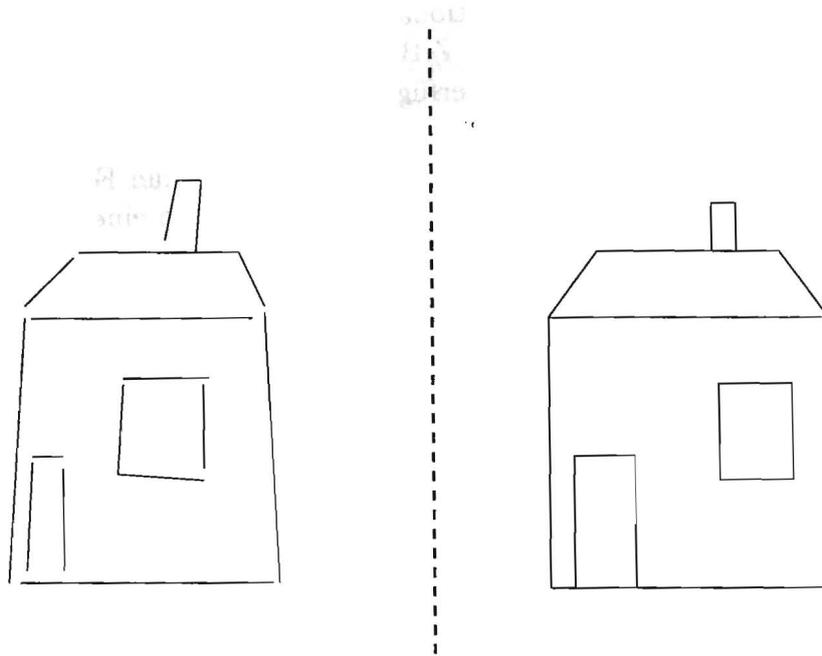


Abbildung 3.13: Verschönerung einer Graphik¹⁴

mit Mühen verbunden und artet oft in anstrengende Kleinarbeit aus. Der Einsatz von Rastern z.B. verspricht hier Hilfe, allerdings erkaufte man sich die genauere Arbeit durch eingeschränkte Platzierungsmöglichkeiten. Besser wäre es, wenn man frei zeichnet, und diese Zeichnung dann quasi per „Knopfdruck“ verschönert. Abbildung 3.13 zeigt das Resultat eines solchen Vorgangs. Ein mit freier Hand gezeichnetes Haus wird korrigiert. Dieses Beispiel weist noch auf ein anderes Einsatzgebiet hin: durch zunehmende Miniaturisierung der Computertechnologie ist die erste Generation von Notepad-Computern¹⁵ serienreif. Diese Computer erkennen handschriftliche Eingaben des Benutzers. Es wäre denkbar, dies auch auf handgemalte Graphiken zu erweitern. In der Arbeit von Bolz [9] wird ein System zur automatischen Verschönerung von handgemalten Graphiken vorgestellt. Bei der Korrektur werden hauptsächlich vorhandene Lücken, die oft beim freihändigen Zeich-

¹⁴Die Abbildung stammt sinngemäß aus [9].

¹⁵Notepad (Englisch) – Notizblock

nen entstehen, geschlossen. Ebenso werden Winkel korrigiert, bzw. Kontraste, wenn nötig, verändert. Dies wird mithilfe einer Datenbank realisiert, die Information über diverse graphische Primitive enthält (wie Kreise und Rechtecke). Hat das System z.B. ein gemaltes Objekt als ein Rechteck identifiziert, so korrigiert es alle Winkel auf 90 Grad. Die Abweichungen müssen dabei innerhalb einer bestimmten Toleranz liegen, um erkannt zu werden.

3.2.5.1 Diskussion

Wenn abstrahierte Darstellungen von Modellen erzeugt werden, die aus Primitiven bestehen (wie z.B. Zylinder, Quader und Laminas), so ist es sinnvoll, jedes Primitiv unterschiedlich im Abstraktionsprozeß zu behandeln, um auf seine geometrischen Eigenschaften konkret einzugehen. Ähnlich wie Bolz dies bei der Verschönerung von Graphiken macht, kann man eine Datenbank erstellen, die wichtige Information über die Abstraktionsmöglichkeiten der Primitiven enthält. Denkbar wäre allerdings auch die Repräsentation dieser Information in prozeduralem Wissen, da dies innerhalb einer objektorientierten Programmierung gut zu realisieren ist.

Kapitel 4

Konzepte eines Systems zur rechnergestützten Abstraktion

Nachdem in Kapitel 2 das Wesen der graphischen Abstraktion und die Mittel untersucht wurden, mit denen Abstraktionen realisiert werden, wendet sich dieses Kapitel dem Problem zu, wie graphische Abstraktionen *rechnergestützt* generiert werden können. Dabei geht es sowohl um die Frage, in welcher Form Eingabedaten repräsentiert werden müssen, als auch darum, wie eine sinnvolle Aufgabenspezifikation für die Abstraktion aussehen sollte. Eine weitere interessante Frage ist, ob und in welcher Form Wissen über die Objekte und den Abstraktionsvorgang benötigt wird. Das Kapitel schließt mit einer Auflistung von Anforderungen, die ein Abstraktionssystem erfüllen sollte.

Basierend auf der Vorarbeit von Feiner (siehe Abschnitt 3.2.1) und den Ausführungen aus Kapitel 2, läßt sich das Grundschema für ein Abstraktionssystem ableiten (siehe Abbildung 4.1). Als Eingabe erhält das System die Abstraktionsaufgabenbeschreibung. Zugleich müssen, da, wie in Abschnitt 2.4 beschrieben, Randbedingungen eine wichtige Rolle im Abstraktionsprozeß spielen, diese dem System bekannt sein.

Das Kernsystem besteht aus einer Komponente für die eigentliche Realisierung der Abstraktion, die eventuell auf Wissen über Abstraktionsvorgänge zurückgreifen muß. Dieser Prozeß basiert auf einer internen Objektrepräsentation, um Veränderungen an den geometrischen Modellen zu ermöglichen. Das dazu benötigte geometrische Wissen steht dem System in Form einer domänenabhängigen externen Wissensquelle zur Verfügung.

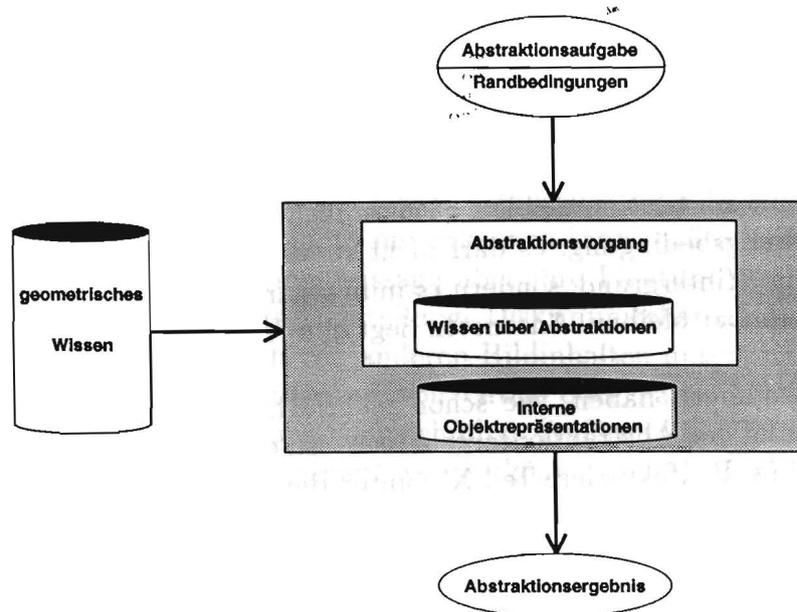


Abbildung 4.1: Grundlegendes Schema für ein Abstraktionssystem.

Schritt für Schritt werden nun die einzelnen Komponenten dieses Grundschemas untersucht und besprochen. Die dabei gewonnenen Erkenntnisse bewirken eine Verfeinerung des Konzeptes, so daß am Ende dieses Kapitels ein detailliertes Schema für ein Abstraktionssystem präsentiert werden kann.

4.1 Die Formulierung der Abstraktionsaufgabe

Die Formulierung der Abstraktionsaufgabe für ein System stößt unter anderem auf die Schwierigkeit, daß die Vorstellung einer Abstraktion schwerfällt. Eine genaue geometrische Zielvorgabe eines Abstraktionsergebnisses erscheint aus diesem Grund nicht sehr sinnvoll. Dies wäre im Einzelfall auch mit viel Mühe verbunden und würde den eigentlichen Abstraktionsvorgang vom System auf den Benutzer übertragen. Das System reduziert sich dann auf eine Bibliothek verschiedener Techniken, ohne selbst eine Auswahl zu

treffen. Alternativ dazu können Forderungen an das Abstraktionsergebnis gestellt werden, die von einem System berücksichtigt werden müssen. Diese Forderungen lassen sich direkt aus den in Abschnitt 2.2 und 2.4 vorgestellten Abstraktionszielen und Randbedingungen einer Abstraktion ableiten.

Möchte man z.B. eine Abstraktion vornehmen, um die Aufmerksamkeit des Betrachters auf ein bestimmtes Detail zu lenken, so unterliegt dieses Detail einer Darstellungsbedingung: es darf nicht so stark abstrahiert werden wie z.B. Details im Hintergrund, sondern es muß sogar noch als dieses bestimmte Detail erkennbar bleiben. Ansonsten liegt eine Verletzung der Zielvorgabe vor.

Randbedingungen haben, wie schon mehrfach erwähnt, großen Einfluß auf die Wirkung des Abstraktionsergebnisses. Eine Idee ist, ein höheres Abstraktionsziel (z. B. „fokussiere Teil X“) unter Berücksichtigung der Randbedingungen in einen Forderungskatalog von Darstellungsbedingungen zu zerlegen. Dies könnte eine vorgeschaltete Planungskomponente realisieren. Die Abstraktionskomponente müßte dann gewährleisten, daß die Bedingungen im Abstraktionsergebnis erfüllt werden.

4.1.1 Darstellungenklassen

Zur Spezifikation des Abstraktionsergebnisses unterscheidet man verschiedene Darstellungenklassen, mit deren Hilfe die Darstellungen der abstrahierten Objekte beschränkt wird. In Anlehnung an [52] werden fünf Darstellungenklassen unterschieden.

1. Sichtbare abstrahierte Darstellungen
2. Wahrnehmbare abstrahierte Darstellungen
3. Diskriminierbare abstrahierte Darstellungen
4. Klassifizierbare abstrahierte Darstellungen
5. Identifizierbare abstrahierte Darstellungen

Die Evaluierung der Zugehörigkeit einer Darstellung zu einer bestimmten Darstellungenklasse wird hier nur unter dem Aspekt der Veränderungen betrachtet, die durch Abstraktion verursacht werden. Weitere Faktoren, die

noch eine Rolle spielen können, sind z.B. die Perspektivenwahl oder der Sehkegel. Für eine umfassende Erläuterung dieser Problematik sei auf [52] verwiesen.

Zur Verdeutlichung der Klassen stelle man sich das Photo (Darstellung) einer bestimmten Person X vor. Ist dieses Photo eine identifizierbare Darstellung von Person X, so ist keine Verwechslung mit irgendeiner Person Y möglich. Wenn man auf dem Photo nur erkennen kann, daß es sich um eine Person handelt, man aber keine Aussage über ihre Identität machen kann, so ist dies eine klassifizierbare Darstellung. Diskriminierbar ist die Darstellung der Person, wenn sie sich von anderen Bildinhalten unterscheiden läßt. Erscheint eine Darstellung aufgrund von Größe oder Farbe besonders augenfällig, so ist diese Darstellung wahrnehmbar. Aufgrund der Tatsache, daß durch Abstraktion Objekte eliminiert werden können, wurde zusätzlich noch die Klasse der sichtbaren abstrahierten Darstellungen eingeführt. Sie enthält alle Darstellungen eines abstrahierten Objekts, das nicht eliminiert wurde. Allgemein gilt: Entsprechend der obigen Nummerierung sind in jeder Darstellungsklasse alle nächst höheren Darstellungsklassen enthalten.

Die Klassen sind in zwei Hauptklassen gruppiert. Die Hauptklasse der **syntaktischen Darstellungen** enthält *sichtbare, wahrnehmbare und diskriminierbare* Darstellungen. Die Klasse der **semantischen Darstellungen** enthält *identifizierbare* und *klassifizierbare* Darstellungen. Syntaktische Darstellungen lassen sich mithilfe einfacher syntaktischer Mittel¹ erkennen. Die semantischen Darstellungen hingegen erfordern zusätzliche Information über den Kontext. Um z.B. zu sagen, daß Person X auf einem Photo identifizierbar ist, benötigt man zunächst die Information, welches bestimmte Kennzeichen dazu geeignet ist (beispielsweise eine Narbe oder die Augenfarbe), und man muß zusätzlich Gewißheit darüber haben, daß Person X keinen Zwillingenbruder oder Doppelgänger hat.

Ohne zusätzliche Information über bestimmte Attribute der einzelnen Objekte kann man keine eindeutige Aussage über die Zugehörigkeit ihrer Darstellungen zu semantischen Klassen machen. Wissen über signifikante Attribute der einzelnen Objekte wird benötigt.

¹Z.B. Kontrastvergleiche oder Schnittbetrachtungen.

4.1.2 Signifikante Attribute

Eine wesentliche Art von Information, die während der automatischen Generierung von Abstraktionen benötigt wird, ist die Information über signifikante Attribute des Modells. Nur anhand dieser Angaben kann, im Hinblick auf die semantischen Klassen, dann eine Abstraktion vorgenommen werden, die eine Spezifikation durch Darstellungsklassen einhält.

In Abschnitt 3.1.1 wurde erläutert, daß es i.a. nicht ausreicht, nur eine Liste von charakteristischen Merkmalen anzugeben, sondern daß zusätzlich eine kontextabhängige Gewichtung dieser Merkmale notwendig ist. Da es unrealistisch ist, für jeden Kontext ein anderes Gewicht zu speichern, muß man eine kontextunabhängige Beschreibung der signifikanten Merkmale finden. Am einfachsten erfolgt diese in der Terminologie der Darstellungsklassen. Dazu wird die Zusammenbauhierarchie des Modells genutzt und die Beschränkung in jedem Vaterknoten durch geeignete Beschränkungen der Söhne definiert. Zur Verdeutlichung betrachte man Abbildung 4.2.

Soll nun die Darstellung des Sportwagens *klassifizierbar* sein, wird diese Forderung durch Bedingungen an die Darstellung der Teile formuliert. So müssen die Räder und die Türen diskriminierbar und der Spoiler klassifizierbar bleiben. Die Idee, die hinter diesem Vorgang steht, ist die, daß man auf diese Weise eine Ableitung von semantischen Darstellungsbedingungen auf syntaktische Darstellungsbedingungen erhält. Diese lassen sich dann von einem Abstraktionssystem verifizieren bzw. einhalten.

Als problematisch erweist sich die Erfassung der signifikanten Attribute. Denn sie lassen sich nicht ohne weiteres nur anhand von syntaktischen Merkmalen ermitteln. Dem Hintergrundwissen über das Objekt kommt oft entscheidende Bedeutung zu. Die Fülle von unterschiedlichen Auffassungen, die man über die Kategorisierung von verschiedenen Objekten haben kann, illustriert [32]. Aus diesen Gründen wird man derzeit um eine manuelle Erstellung der Datenbank nicht herumkommen.

Die in diesem Abschnitt gewonnenen Erkenntnisse sind in Abbildung 4.3 zusammengefaßt. Die Randbedingungen werden in die Aufgabenstellung integriert, und die externe geometrische Wissensbasis besteht aus den Repräsentationen der Objekte und der Information über signifikante Attribute.

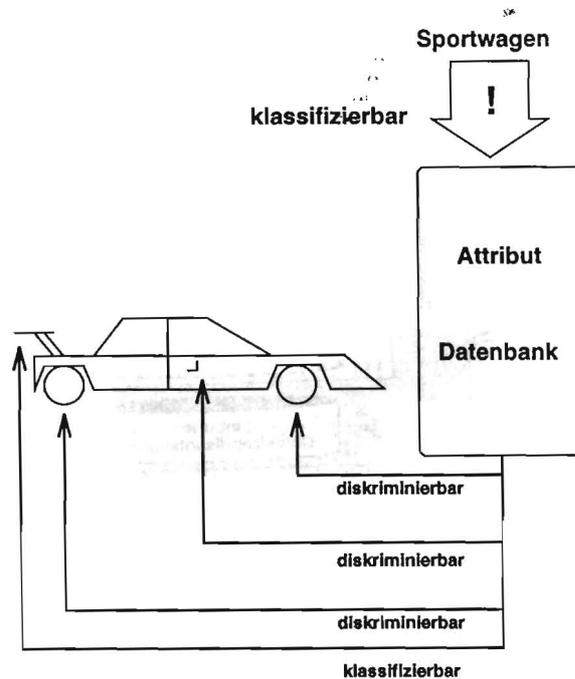


Abbildung 4.2: Signifikante Attribute eines Sportwagen.

4.2 Repräsentation der graphischen Elemente

Die Abstraktionen, die in Kapitel 2 Gegenstand der Untersuchung waren, stammten zum größten Teil aus technischen Dokumentationen und Bedienungsanleitungen. Diese Dokumente liegen in den meisten Fällen in Papierform vor und sind damit zweidimensionaler Natur. Es stellt sich hier die Frage, welche Art von Bildrepräsentationen verwendet werden sollte. Im System APEX (siehe auch Kapitel 3) werden ausschließlich dreidimensionale Objektrepräsentationen verwendet. Es ist aber, aus oben genannten Gründen, naheliegend, Abstraktionen auch auf zweidimensionalen Bildrepräsentationen durchzuführen (zu bedenken bleibt, daß dreidimensionale Objekte wesentlich mehr Information beinhalten als zweidimensionale). Zudem besteht bei der Erstellung von Graphiken ein Zusammenhang zwischen dreidimensio-

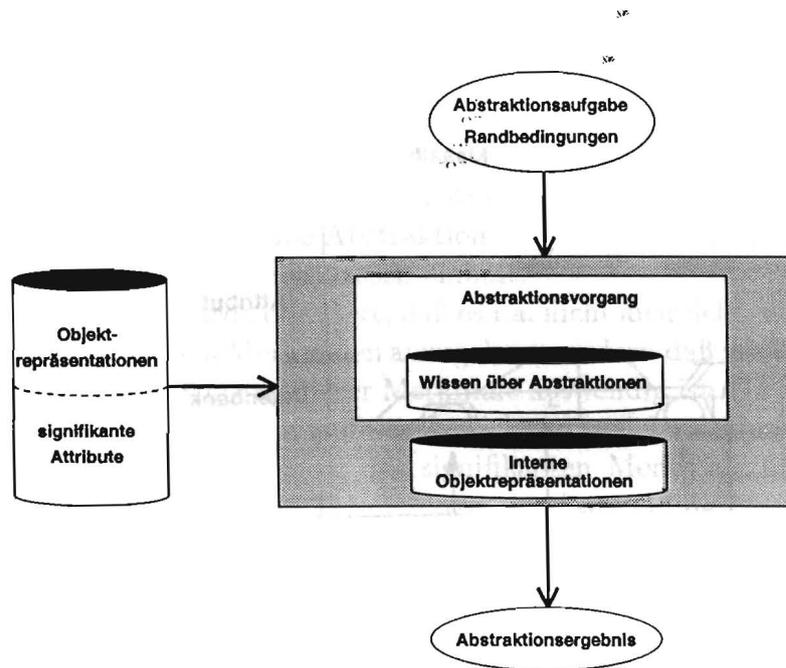


Abbildung 4.3: Signifikante Attribute müssen berücksichtigt werden.

nalen und zweidimensionalen Objekten: die zweidimensionalen Bildobjekte entstehen aus den dreidimensionalen Objekten durch eine Projektion. Diese Projektion wird entweder von einem Graphiker ausgeführt (in diesem Fall besitzt der Graphiker eine dreidimensionale Vorstellung des Objekts, das er gerne darstellen möchte), oder durch ein Graphiksystem mithilfe von Projektionsalgorithmen² berechnet. Diese Tatsache unterstützt die Abstraktion auf dreidimensionalen Datenstrukturen. Denn Abstraktionen auf diesen Strukturen bleiben auch nach einer weiteren Projektion ins Zweidimensionale erhalten.

4.2.1 Externe Objektrepräsentation

An die geometrische Beschreibung der benutzten Domäne werden besondere Anforderungen gestellt. Um Abstraktionen an dreidimensionalen Objekten vornehmen zu können, wird Information bezüglich der Volumina, der Flächen

²Z.B. perspektivische oder parallele Projektion.

und ihrer Strukturen benötigt. Darüber hinaus muß eine Zusammenbauinformation existieren, die die Gruppierung einzelner Objektteile erklärt. Eine sehr allgemeine Modellrepräsentation, die diese Forderungen erfüllt, ist das *annotierte Drahtrahmenmodell*. Dieses Modell enthält sowohl eine Information über die Kanten und Flächen als auch (annotierte) Information über Volumen, Struktur und Farbe. Einzelne Objekte können schrittweise zu funktionalen Einheiten gruppiert werden. Dieser Vorgang definiert dann eine Zusammenbauhierarchie.

4.2.2 Interne Objektrepräsentation

Eine interne Kopie der externen Objektrepräsentation erlaubt eine gefahrlose Modifikation des 3D-Modells. Darüber hinaus bietet es sich an, wie im System APEX beschrieben (siehe Abschnitt 3.2.1), eine interne Abstraktionshierarchie zu benutzen, um so später auf verschiedene Abstraktionsgrade zurückzugreifen. Eine interne zweidimensionale Objektrepräsentation wird benötigt, um auch zweidimensionale Abstraktionen durchzuführen. Dabei müssen Bildinhalte so repräsentiert werden, daß Überlegungen zu den Formen bzw. Silhouetten, Farben und Strichstärken möglich sind. Eine reine Bitmapdarstellung der Bildobjekte ist daher völlig unzureichend. Günstiger ist ein Ansatz, der für jedes Objekt im Bild eine Kantenliste enthält. Jeder Kante entsprechen zwei Bildschirmkoordinatenpaare. Zusätzlich wird jeder Kante eine Strichstärke und jedem Objekt eine Farbe zugeordnet.

4.3 Der Abstraktionsvorgang

Bei der Besprechung der Objektrepräsentationen ist deutlich geworden, daß es zwei prinzipielle Ebenen gibt auf denen Abstraktionen vorgenommen werden können. Wird über die dreidimensionalen Objektstrukturen (3D-Modelle) abstrahiert, so ist dies eine Abstraktion auf der *Modellebene*. Abstraktionen mithilfe der zweidimensionalen Repräsentationen finden auf der *Projektionsebene* statt. So spielen bei der Abstraktion auf der *Modellebene* die Lage der Objekte im Raum und ihre Dimensionen eine wichtige Rolle, während bei der Abstraktion auf der *Projektionsebene* die Beschaffenheit der Silhouetten und Kanten im Vordergrund steht. Dies gilt es bei der Wahl der Abstraktionstypen zu berücksichtigen, denn sie eignen sich unterschiedlich

Modellebene	Projektionsebene
Verschmelzen, Skalieren, Weglassen, Dimensionieren	Vereinheitlichen, Weglassen, Symbolisieren

Tabelle 4.1: Zusammenhang zwischen Ebenen und Abstraktionstypen.

gut für die verschiedenen Abstraktionsebenen. Tabelle 4.1 ordnet den Ebenen die geeigneten Abstraktionstypen zu.

Eine wichtige Leistung, die während des Abstraktionsvorgangs von einem Abstraktionssystem vollbracht werden muß, ist die Einhaltung der Darstellungsbedingungen im Abstraktionsergebnis. Dazu benötigt das System Wissen über allgemeine Abstraktionsvorgänge, um den jeweils geeigneten Abstraktionstyp (siehe Abschnitt 2.3) richtig zu verwenden. Diese Auswirkungen können auf den verschiedenen Abstraktionsebenen unterschiedlich ausfallen. Daher muß sowohl die *Modellebene* als auch die *Projektionsebene* gesondert behandelt werden. Im nächsten Kapitel wird ein Vorschlag gemacht, welche Form dieses Wissen hat und wie es verarbeitet wird.

4.4 Forderungen an ein Abstraktionssystem

Der vorhergehende Abschnitt hat das Schema eines Abstraktionssystems weiter vervollständigt (siehe Abbildung 4.4). Der Abstraktionsvorgang arbeitet sowohl auf der *Projektionsebene*, als auch auf der *Modellebene*. Dementsprechend zweigeteilt ist die interne Wissensbasis. Die interne Objektrepräsentation teilt sich in eine 3D- und eine 2D-Repräsentation, und auch das Wissen über den Abstraktionsvorgang unterteilt sich in den Teil auf der *Modellebene* und in den Teil auf der *Projektionsebene*. Aus diesem Konzept und den Untersuchungen, die in den letzten Kapiteln gemacht wurden, läßt sich ein Forderungskatalog ableiten, den ein Abstraktionssystem erfüllen sollte:

- Spezifizierung des Abstraktionsergebnisses durch Darstellungsbedingungen.

- Verwendung von Wissen über die Objektmodelle (signifikante Attribute).
- Verwendung der wichtigsten Abstraktionstypen.
- Wissen über die Auswirkungen der Abstraktionstypen auf das Abstraktionsergebnis muß zur Verfügung stehen.
- Abstraktionen sollten sowohl auf der Modell- als auch auf der Projektionsebene möglich sein.

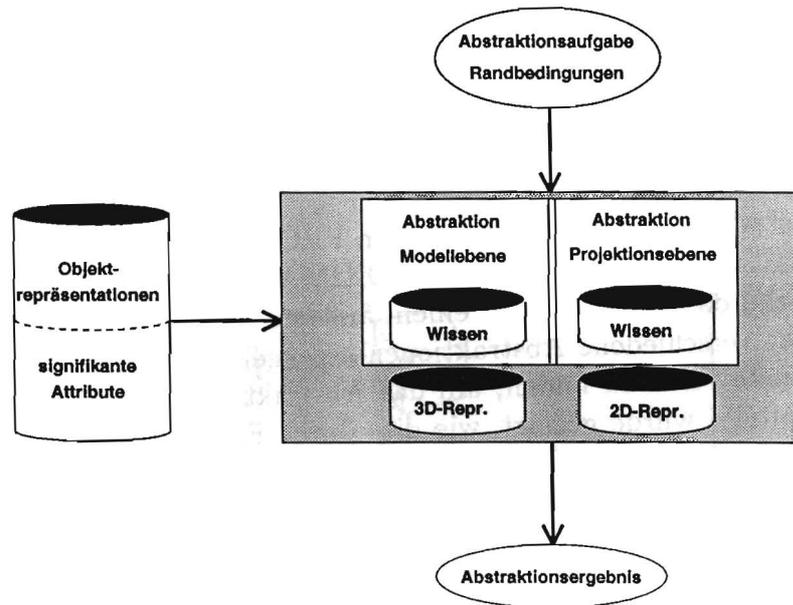


Abbildung 4.4: Das vervollständigte Abstraktionsschema.

Kapitel 5

PROXIMA: Ein Werkzeug zur Erzeugung graphischer Abstraktionen

In diesem Kapitel wird das Abstraktionssystem PROXIMA¹ vorgestellt. Ein wichtiger Grundsatz der Konzeption von PROXIMA bestand darin, ein System zu entwickeln, das abhängig von einem äußeren Kontext von ein und demselben Objekt verschiedene Abstraktionen erzeugen kann. Dazu war es nötig, eine geeignete Form zu finden, um das Abstraktionsergebnis zu spezifizieren. In Kapitel 4 wurde gezeigt, wie dies durch Formulieren von Darstellungsbedingungen geschehen kann. Höhere Abstraktionsziele, wie sie in Abschnitt 2.2 vorgestellt wurden, können durch Zerlegung in Darstellungsbedingungen formuliert werden.

PROXIMA erhält als Eingabe das geometrische Wissen über die Abstraktionsdomäne und die bereits erwähnten Darstellungsbedingungen an das Abstraktionsergebnis. In einem iterativen Abstraktionsvorgang wird eine Abstraktion vorgenommen, in der die Darstellungsbedingungen eingehalten werden.

In den folgenden Abschnitten stehen zunächst die geometrischen Wissensstrukturen im Vordergrund, mit denen PROXIMA arbeitet. Daran anschließend wird erläutert, welche Abstraktionsziele bzw. -typen verwendet werden, und es wird schließlich ausführlich der eigentliche Abstraktionsvor-

¹PROXIMA ist ein Wortfragment von *Approximation*.

gang beschrieben.

5.1 Geometrische Wissensstrukturen

PROXIMA benötigt geometrische Repräsentationen der Objekte und zudem Information über signifikante Attribute.

5.1.1 Objektrepräsentationen

In Abschnitt 4.2 wurde erläutert, welche Anforderungen an die interne und externe Objektrepräsentation gestellt werden.

5.1.1.1 Externe Objektrepräsentation

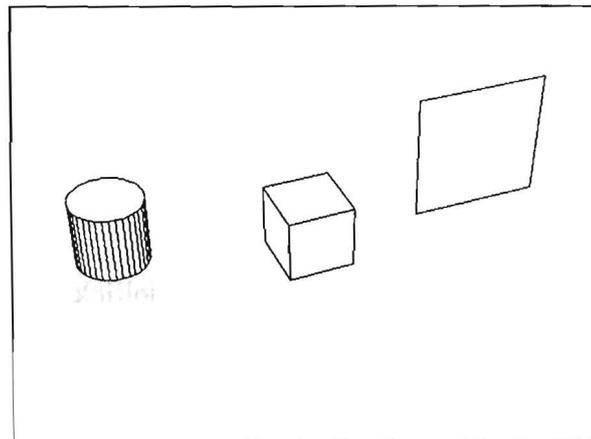


Abbildung 5.1: Die drei Grundprimitive: Zylinder, Quader und Lamina.

PROXIMA arbeitet mit annotierten Drahtrahmenmodellen. Diese Modelle erlauben es, Schlüsse über die Lage der Objekte im Raum, ihre Struktur, ihre Farbe und ihre Volumina zu ziehen. Die Modelle werden aus drei Grundprimitiven zusammengesetzt. Diese sind Zylinder, Quader und Lamina (siehe Abbildung 5.1). Zusätzlich können aus dem Abstraktionsprozeß

auch deformierte Quader oder Zylinder² hervorgehen (siehe Abbildung 5.5). Die räumlichen Ausdehnungen der Objektprimitiven werden durch ihre Seitenflächen begrenzt. Quader bestehen aus sechs Flächen. Zylinder bestehen aus 32 rechteckigen Seitenflächen und jeweils einer Ober- und Unterfläche und Lamina aus einer jeweils rechteckigen Vorder- und Rückfläche. Durch Zusammensetzen dieser Grundprimitive in variablen Größen entstehen komplexere Objekte. Dabei sind auch Drehungen der Primitive um beliebige Winkel erlaubt. Die möglichen Modelldomänen beschränken sich also nicht auf eine koordinatenachsenparallele Welt³.

Ein komplexes Objekt wird durch seine Zusammenbauhierarchie repräsentiert. Dies ist ein Baum, dessen innere Knoten zusammengesetzte Objekte enthalten und dessen Blätter aus Grundprimitiven bestehen. Eine Kante im Baum kennzeichnet den Zusammenbau der Sohnmodelle zum Vatermodell. Eine oder mehrere solcher Zusammenbauhierarchien bilden zusammen eine Objektkonstellation. Eine Bildszene entsteht aus dieser Objektkonstellation, wenn jedem Objekt ein Sichtbarkeitsattribut hinzugefügt wird. Eine Bildszene enthält also nur die für einen Betrachter sichtbaren Objekte.

5.1.1.2 Interne Objektrepräsentation

In Anlehnung an [20] (siehe auch den Abschnitt 3.2.1 über APEX) baut PROXIMA zu der Zusammenbauhierarchie eine parallele Struktur (die Abstraktionsobjekthierarchie) dreidimensionaler Abstraktionsobjekte auf. Jedes Abstraktionsobjekt enthält einen Verweis auf das jeweilige annotierte Drahtrahmenmodell und zusätzliche Attribute, die während des Abstraktionsprozesses von Bedeutung sind (wie z.B. die Information über die Darstellungsbedingungen des Modells). Im Laufe des Abstraktionsvorgangs auf der Modellebene wird nur mit der Abstraktionsobjekthierarchie gearbeitet. Die ursprüngliche Zusammenbauhierarchie bleibt unberührt.

Für die zweidimensionale Abstraktion auf der Projektionsebene wird eine Struktur verwendet, die zu jedem dreidimensionalen Objekt seine Kantenrepräsentation in Bildschirmkoordinaten enthält. Dies kann natürlich nur in Abhängigkeit einer bestimmten Perspektive geschehen. Das Problem der geeigneten Perspektivenwahl ist nicht trivial und wird in [49] ausführlich be-

²Wenn im folgenden von Quadern oder Zylindern die Rede ist, sind die deformierten Versionen miteinbegriffen.

³Wie z.B. bei APEX [20].

handelt. PROXIMA benutzt entweder eine Standardperspektive oder erlaubt dem Benutzer, diese interaktiv zu ändern.

5.1.2 Signifikante Attribute

Abstraktionsergebnisse werden durch die Beschränkung der Darstellungen der Objektteile spezifiziert. Wie in Abschnitt 4.1.2 erläutert, wird hierzu Wissen bezüglich der signifikanten Attribute des Objektmodells benötigt⁴. Ideal wäre es, diese Daten automatisch zu gewinnen. Leider ist dies aus den in Abschnitt 4.1.2 genannten Gründen nicht möglich. Daher greift PROXIMA auf eine Datenbank zu, in der dieses Wissen gespeichert ist.

Für ein Objektteil besteht für jede der fünf Darstellungsbedingungen (zur Erinnerung: sichtbar, wahrnehmbar, diskriminierbar, klassifizierbar und identifizierbar) ein Eintrag, der festlegt, wie sich diese Darstellungsbedingung auf die Darstellungsbedingungen der Söhne in der Zusammenbauhierarchie auswirkt.

Betrachtet man z.B. die IC-Bank in Abbildung 5.2, so ergeben sich aus der Darstellungsbedingung *klassifizierbar*, *diskriminierbare* Darstellungsbedingungen der vier ICs⁵.

Für jedes zusammengesetzte Objekt in der Zusammenbauhierarchie muß eine solche Tabelleninformation gespeichert werden. Wären z.B. die einzelnen ICs zusammengesetzte Objekte, so gäbe es auch für sie jeweils eine Tabelle. Wird nun ein Objekt durch eine Darstellungsbedingung beschränkt, so können durch eine rekursive Abfrage der Datenbank alle Teilobjekte ebenfalls beschränkt werden.

5.2 Abstraktionsziele und Abstraktionstypen

In Abschnitt 2.2 wurde eine Reihe von Abstraktionszielen beschrieben, die durch Abstraktion erfüllt werden können. Damit PROXIMA einen Abstraktionsvorgang starten kann, müssen diese höheren Ziele in Darstellungsbe-

⁴Ähnliches Wissen wird z.B. auch bei der „Verschönerung von Graphiken“ benötigt (siehe dazu auch Abschnitt 3.2.5).

⁵Vergleiche dazu die Tabelle in Abbildung 5.2.

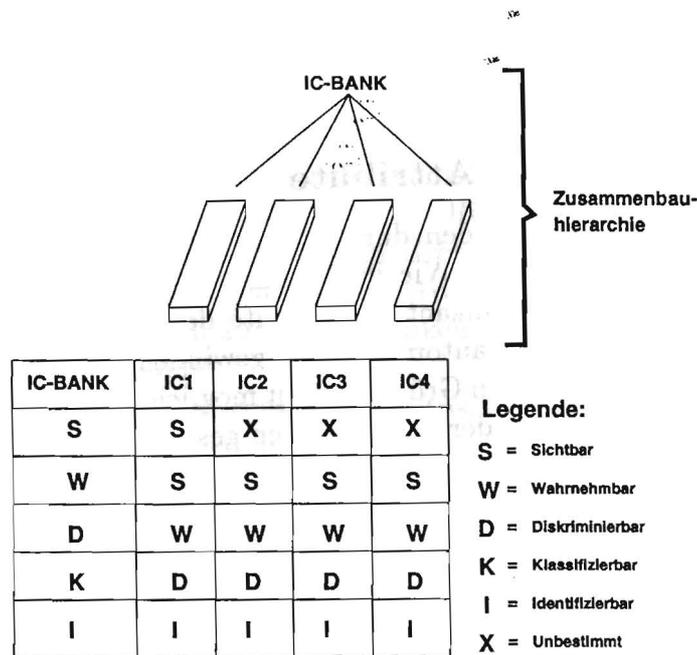


Abbildung 5.2: Die Auswirkungen der Darstellungsbedingungen der IC-Bank auf die Darstellungsbedingungen der vier ICs.

dingungen an die geometrischen Objekte formuliert werden. Diese Zerlegung ist auch abhängig von den in Abschnitt 2.4 erläuterten Randbedingungen (dem situativen Kontext und den Persistenz- und Konsistenzbedingungen). Vorschläge für eine solche Zerlegung werden später in Kapitel 6 gemacht, indem einige Beispielabstraktionen vorgestellt werden.

PROXIMA verwendet die wichtigsten in Abschnitt 2.3 vorgestellten Abstraktionstypen (vergl. dazu auch die Tabelle in Abschnitt 2.5). Dies sind auf der Modellebene die Abstraktionstypen *Verschmelzen*, *Dimensionieren* und *Weglassen*, sowie auf der Projektionsebene die Abstraktionstypen *Vereinheitlichen* und *Verändern der Strichstärken*. Diese Abstraktionstypen sind auch am besten in der Lage, gestaltpsychologische und designrelevante Grundsätze zu berücksichtigen, wie sie in Abschnitt 3.1.2 besprochen wurden.

Das *Dimensionieren* und *Verschmelzen* sind zwei Abstraktionstypen, die eng miteinander verwandt sind. Dies veranschaulichen die Abbildungen 2.3 und 2.4 in Abschnitt 2.3. Das *Dimensionieren* kann als eine Vorstufe zum

Verschmelzen betrachtet werden. Denn im ersten Abstraktionsschritt „wachsen“ die beiden gegenüberliegenden Flächen aufeinander zu und verändern dabei die Dimensionen der Objekte. Und im nächsten Schritt verschmelzen die beiden Objekte zu einem neuen Gesamtobjekt. Am Beispiel einer LED-Reihe wird dies in *Abbildung 5.3* verdeutlicht.

Beide Abstraktionstypen werden von PROXIMA als binäre Operatoren verwendet. Dies bedeutet, daß stets zwei Objekte betroffen sind. Beim *Dimensionieren* „wachsen“ immer Flächen von zwei Objekten aufeinander zu und beim *Verschmelzen* werden ebenfalls immer zwei Objekte zu einem neuen zusammengefaßt.

Auf der Projektionsebene werden die Abstraktionstypen *Vereinheitlichen* und *Verändern der Strichstärken* verwendet. Das Vereinheitlichen mehrerer Objekte gelingt, indem ihre Objektkanten „gestrichelt“ werden. PROXIMA kann dabei die Strichstärken und Strichabstände variieren. Beide Abstraktionstypen der Projektionsebene werden vom System als unäre Operatoren eingesetzt.

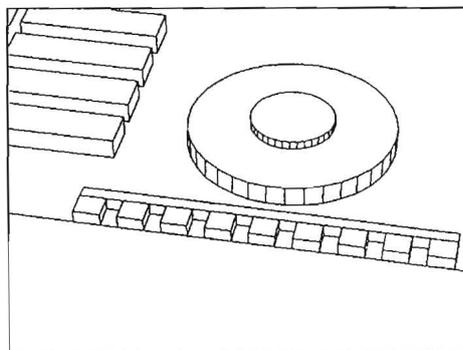


Abb.5.3 (a): Die Ausgangslage.

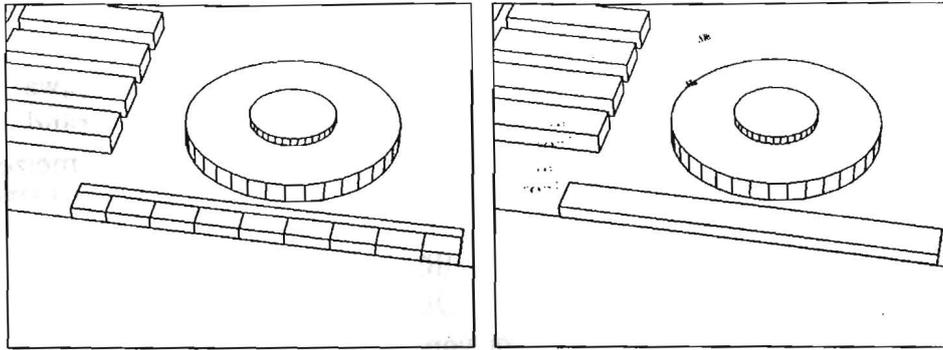


Abb.5.3 (b): Die einzelnen LEDs „wachsen“ aufeinander zu...

Abb.5.3 (c): ...und „verschmelzen“ dann in einem zweiten Schritt.

5.3 Die Formulierung der Abstraktionsaufgabe

Den Kern der Abstraktionsspezifikation bildet eine Liste von Darstellungsbedingungen an die Objekte einer Bildszene. Zusätzlich kann PROXIMA noch mitgeteilt werden, ob Bildobjekte im Vorder- oder im Hintergrund liegen. Ein weiterer Steuermechanismus steht durch die Variation des allgemeinen Abstraktionsgrads zur Verfügung. Die in Kapitel 4 besprochenen Schwierigkeiten, die sich bei der Vorstellung von Abstraktionen ergeben, werden durch das Erhöhen oder das Erniedrigen dieses Parameters ausgeglichen. Eine Abstraktionsaufgabe⁶ für PROXIMA könnte dementsprechend folgendermaßen aussehen:

⁶Die Syntax dieses Beispiels orientiert sich an der LISP-Notation.

```

(Abstrahiere-Objekt 'Modem-Platine
  (Darstellungsbedingung 'Schalter4 'identifizierbar)
  (Darstellungsbedingung 'IC-Bank 'klassifizierbar)
  (Darstellungsbedingung 'RS232 'sichtbar)
  (Darstellungsbedingung 'Transformator 'diskriminierbar)
  (Vordergrundobjekte 'Schalter4 & IC-Bank)
  (Hintergrundobjekte 'Rest)
  (Abstraktionsniveau 'Mittel)
)

```

Hier soll das Modell einer Modemplatine abstrahiert werden. Im abstrahierten Ergebnis soll der *Schalter4* identifizierbar, die *IC-Bank* klassifizierbar, die *RS232-Schnittstelle* sichtbar und der *Transformator* diskriminierbar bleiben. Außerdem wird der *Schalter4* als Vordergrundobjekt deklariert und ein mittleres Abstraktionsniveau gewählt.

Wie PROXIMA nun aus einer solchen Vorgabe eine Abstraktion realisiert, wird im nächsten Abschnitt behandelt.

5.4 Der Abstraktionsvorgang

PROXIMA erhält eine Spezifikation des Abstraktionsergebnisses, entweder von einer übergeordneten Planungskomponente, oder aber durch die direkte Eingabe seitens des Benutzers. Dann beginnt der Abstraktionsvorgang, indem der lokale Kontext gesetzt wird. Das bedeutet, daß für alle Objektteile die noch fehlenden Darstellungsbedingungen durch ein Propagierungsverfahren ermittelt werden. Ist dieser Prozeß abgeschlossen, beginnt der eigentliche Abstraktionsvorgang. Dabei wird zunächst auf der Modellebene (3D-Abstraktion) abstrahiert und dann auf der Projektionsebene (2D-Abstraktion).

Die Abstraktion der Modelldaten erfolgt durch eine Schleife, in der der Abstraktionsgrad schrittweise angehoben wird, bis die Darstellungsbedingungen ausgereizt sind. Dies wird durch die verwendeten Abstraktionstypen

Dimensionieren und *Verschmelzen* begünstigt, da in einem Schritt dimensioniert und im nächsten verschmolzen wird. Unabhängig davon erfolgt die Abstraktion auf der Projektionsebene. Sie ist bereits durch den lokalen Kontext festgelegt und kann entweder nach der Abstraktion auf der Modellebene oder auch direkt ausgeführt werden. Der nachfolgende Pseudocode verdeutlicht die prinzipielle Arbeitsweise von PROXIMA.

```
PROC Abstraktionsprozeß
  MODULE Setze lokalen Kontext
  IF Abstraktion auf Modellebene?
    THEN
      LOOP
        MODULE Erhöhe Abstraktionsgrad auf der Modellebene
        UNTIL Darstellungsbedingungen erreicht
      POOL
    FI
  IF Abstraktion auf Projektionsebene?
    THEN
      MODULE Abstrahiere auf der Projektionsebene
    FI
CORP
```

5.4.1 Der lokale Kontext

Zu Beginn der eigentlichen Abstraktion wird zunächst der lokale Kontext gesetzt. Unter dem lokalen Kontext versteht man den Teil der internen Objektrepräsentation, der die Information über die Darstellungsbedingungen enthält. Da die Abstraktionsaufgabe nur für einige Objekte konkrete Darstellungsbedingungen festlegt, ist der lokale Kontext zu Beginn noch unvollständig.

Unter Berücksichtigung der Ausführungen in Abschnitt 4.1.2 führt PROXIMA in dieser Phase drei Schritte durch:

1. Es wird für jedes Objektteil, das nicht direkt spezifiziert wurde, eine Darstellungsbedingung ermittelt, und eine eventuelle Kollision von mehreren Darstellungsbedingungen für ein Objektteil wird aufgelöst.
2. Es werden Standardeinstellungen ermittelt.

3. Die Abstraktionsmethode für die Abstraktion auf der Projektionsebene wird festgelegt, da diese in PROXIMA ausschließlich auf dem lokalen Kontext basiert.

Für jede Modelldomäne existiert eine Attributdatenbank, in der, basierend auf der Zusammenbauhierarchie des Modells, gespeichert ist, wie sich Darstellungsbedingungen der zusammengesetzten Objekte auf ihre Einzelteile auswirken. Diese Information (über signifikante Attribute) wird genutzt, um jedes Objektteil einer Darstellungsbedingung zu unterwerfen. Erhält PROXIMA eine Spezifikation für ein Objekt, so kann mithilfe der Attributdatenbank jedem seiner Teilobjekte eine Bedingung auferlegt werden. Zur Verdeutlichung betrachte man das Beispiel in Abbildung 5.4. Ein Objekt O besteht aus den Teilobjekten O_1 , O_2 und O_3 . Diese bestehen wiederum aus den Teilen P_{11} , P_{12} , P_{21} , P_{22} , P_{31} , P_{32} und P_{33} . Die Datenbankeinträge für jedes Objekt werden durch die entsprechenden Tabellen dargestellt. Für jeden inneren Knoten der Zusammenbauhierarchie (inklusive der Wurzel) existiert ein solcher Eintrag. Die Abstraktionsaufgabe lautet: abstrahiere Objekt O unter den folgenden Bedingungen:

1. Die abstrahierte Darstellung des Gesamtobjekts O soll wahrnehmbar bleiben.
2. Objekt O_1 soll im Vordergrund liegen.
3. Objekt O_2 bleibt diskriminierbar.
4. Objekt O_3 bleibt sichtbar.

Der Baum in der Abbildung 5.4 verdeutlicht die Auswirkung der Beschränkungen der Objekte O_1 , O_2 und O_3 auf die Teilobjekte P_{1n} , P_{2n} und P_{3n} . Die durch Kreise hervorgehobenen Darstellungsbedingungen stammen direkt aus der Aufgabenspezifikation. Alle anderen Darstellungsbedingungen werden durch die entsprechenden Tabellen ermittelt.

Während dieser Propagierung können Konflikte entstehen, wenn für ein Objektteil mehrere Darstellungsbedingungen vorliegen. In dem obigen Beispiel ist dies beim Objekt O_1 der Fall. Dieses Objekt liegt im Vordergrund und muß damit identifizierbar bleiben. Gleichzeitig erhält es aber eine Beschränkung auf eine sichtbare Darstellung über das Vaterobjekt O , entsprechend dem Eintrag in der Tabelle für O , unter wahrnehmbar (w). Im Falle ei-

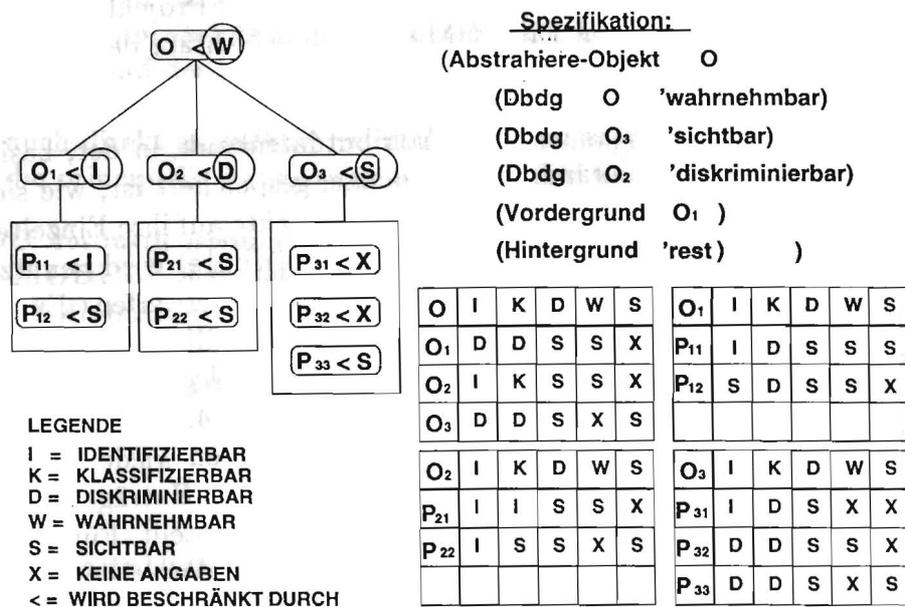


Abbildung 5.4: Wie wirkt sich die Spezifikation auf den lokalen Kontext aus?

nes solchen Konflikts behauptet sich die „stärkere“ Darstellungsbedingung⁷. Also bleibt die Darstellungsbedingung *sichtbar* unberücksichtigt.

Liegen PROXIMA für ein Teilobjekt keine Beschränkungen vor, so versucht das System eine Defaultbeschränkung zu finden. Diese wird in Abhängigkeit des allgemeinen Abstraktionsniveaus bzw. der Vorder- und Hintergrundobjekte gewählt.

Liegen nun für alle Objektteile Darstellungsbedingungen vor, wird die genaue Vorgehensweise für die Abstraktion auf der Projektionsebene festgelegt. Dabei wird jeder Kante eines Objekts eine Strichstärke und ein Strichelfaktor⁸ zugewiesen. Auch dies geschieht in Abhängigkeit davon, ob ein Objekt im Vorder- oder Hintergrund liegt, bzw. welcher Darstellungsbeschränkung es unterworfen wurde.

⁷Dabei ist die Reihenfolge ganz intuitiv: 1. identifizierbar, 2. klassifizierbar, 3. diskriminierbar, 4. wahrnehmbar, 5. sichtbar.

⁸Der Strichelfaktor legt fest, wie groß die Lücken zwischen den Strichen sein sollen, also wie stark gestrichelt wird.

Die Projektionsabstraktion kann nun direkt ausgeführt werden. Normalerweise wird jedoch zunächst die Modellabstraktion durchgeführt, denn sie erlaubt eine variabelere Gestaltung des Abstraktionsresultats.

5.4.2 Die Modellabstraktion

3D-Modelle abstrahiert PROXIMA mithilfe der Abstraktionstypen *Dimensionieren*, *Verschmelzen* und *Weglassen*. Dabei ist das *Weglassen* technisch am einfachsten zu realisieren, da es genügt, das entsprechende Objekt aus der Bildszene zu entfernen. Die Realisierung der beiden anderen Abstraktionstypen gestaltet sich schwieriger. Sowohl das *Verschmelzen*, als auch das *Dimensionieren* sind keine eindeutigen Operationen. Viele verschiedene Ergebnisse sind bei gleicher Objektkonstellation denkbar. Eine Parametrisierung der Abstraktionstypen muß diesen Umständen Rechnung tragen.

Eine weitere wichtige Information, die zur Modellabstraktion benötigt wird, findet sich auch schon bei Feiner (siehe Abschnitt 3.2.1). Sie betrifft die räumliche Lage der Objekte zueinander und ihre Position in der Zusammenbauhierarchie. Letztere erlaubt eine Zuordnung des Objekts zu einer funktionalen Gruppe. Kennt man die räumlichen Relationen, so lassen sich Objektgruppen im Raum bilden, die aus bildgestalterischen Gesichtspunkten (siehe Abschnitt 3.1.2) wichtig sind. Umfangreiche Gruppierungen ergeben stärkere Abstraktionen.

Die räumlichen Relationen lassen sich zunächst nicht direkt aus der Zusammenbauhierarchie ableiten. Sie müssen in einer ersten Stufe errechnet werden. Basierend auf den dort gewonnenen Erkenntnissen werden dann in einer zweiten Stufe die Abstraktionen erzeugt.

5.4.2.1 Berechnung der räumlichen Relationen

Die Ausdehnungen der Grundprimitiven (Quader, Zylinder und Lamina) werden durch ihre Seitenflächen bestimmt. Jeder dieser Seiten wird die nächstliegende Seite eines benachbarten Objekts zugeordnet. Bei der Betrachtung des Zylinders werden dabei nicht alle 32 Seitenflächen berücksichtigt, sondern das Verfahren beschränkt sich auf sechs Referenzflächen. Wenn die Anzahl der Referenzflächen erhöht wird, so hat dies einen erheblichen Effizienzverlust zur Folge, der nicht durch bessere Resultate aufgewogen wird.

Zwei Seiten werden dann als nächstliegend angesehen, wenn folgende Bedingungen erfüllt sind:

1. Der Schnitt des Polygonzugs, der durch die Kanten der einen Fläche bestimmt wird, und des Polygonzugs, der durch die orthogonale Projektion der anderen Fläche in die Ebene der ersten Fläche entsteht, wird für jede der beiden Flächen betrachtet. Ist einer der beiden Schnitte nicht leer, so gelten die beiden Flächen als *benachbart*.
2. Der Polygonzug, der durch den Schnitt entsteht, muß eine Fläche begrenzen. Dies soll verhindern, daß z.B. zwei im 90 Grad Winkel zueinanderstehende Flächen als nächstliegend angesehen werden.
3. Es gibt zu beiden Flächen keine näherliegenden benachbarten Flächen. Die Nähe zweier Flächen wird dabei über ihre mittlere Entfernung bestimmt.

Der einfachste Algorithmus zur Lösung dieses Problem berechnet für alle Flächenpaare die Entfernungen und vergleicht diese miteinander. Für jedes Grundprimitiv müssen sechs Flächen verglichen werden. Dies führt bei n Primitiven zu $\frac{1}{2}(n^2 - n)$ Vergleichen. In jedem Vergleich müssen 6^2 Flächen berücksichtigt werden. Insgesamt führt dies zu $\frac{1}{2}6^2(n^2 - n)$ Flächenvergleichen. Jeder Vergleich beinhaltet aufwendige geometrische Berechnungen. Bereits bei einfacher Modellierung der Domänen (ca. 50 - 100 Grundprimitive) führt dies auf der verwendeten Rechenanlage⁹ zu Laufzeiten im Minutenbereich. Um dieses Problem zu entschärfen, greift der modifizierte Algorithmus auf Heuristiken zurück. Es werden untereinander nur Objekte verglichen, die durch die Zusammenbauhierarchie gruppiert wurden. Besteht z.B. eine IC-Bank aus vier ICs, so wird die nächstliegende benachbarte Fläche eines ICs nur bei den drei anderen Elementen der IC-Bank gesucht. Des weiteren wird der Suchraum innerhalb der Gruppe beschränkt. Dies geschieht, indem, ausgehend von einer Fläche, nur in einem bestimmten Radius gesucht wird. Wird keine benachbarte Fläche innerhalb dieses Radius gefunden, so wird dieser Radius schrittweise vergrößert, bis er alle Objekte der Gruppe umfaßt.

⁹XL1200 Symbolics Lispmaschine

5.4.2.2 Parametrisierung der Abstraktionstypen

Vielfältige Abstraktionsergebnisse lassen sich erst realisieren, wenn die entsprechenden Operatoren variabel zu handhaben sind. Daher bietet es sich an, die Abstraktionstypen *Dimensionieren* und *Verschmelzen* zu parametrisieren. Während des Abstraktionsvorgangs kann dann jeder dieser Abstraktionstypen, angepaßt an die jeweilige Situation, eingesetzt werden.

Beim *Dimensionieren* wird die Größe eines Objekts verändert, indem eine oder mehrere seiner Flächen verschoben werden. Da das *Dimensionieren* eine binäre Operation ist, werden dabei immer Lücken zwischen Objekten geschlossen. Die Größe dieser Lücke entspricht also dem maximalen Wert, um den eine der beiden Fläche verschoben werden kann.

Sei nun Δ der Abstand zwischen den beiden Objekten und $\lambda \in [0 \dots 1]$, dann wird einer Fläche der Wachstumswert

$$\Phi_{Fläche1} := \lambda * \Delta$$

zugewiesen. Die andere Fläche wird um den Wert

$$\Phi_{Fläche2} := (1 - \lambda) * \Delta$$

verschoben.

Durch Veränderung des Parameters λ wird also der „Berührungspunkt“ der beiden Flächen beeinflusst. Dies führt zu unterschiedliche Abstraktionsergebnissen. Zum Beispiel kann λ abhängig von den Massen der Objekte gewählt werden. Sind \mathcal{A}, \mathcal{B} die Massen der Objekte so gilt:

$$\lambda = \frac{\mathcal{A}}{\mathcal{A} + \mathcal{B}}.$$

Die Abbildung 5.5 illustriert verschiedene Auswirkungen der Parameterwahl.

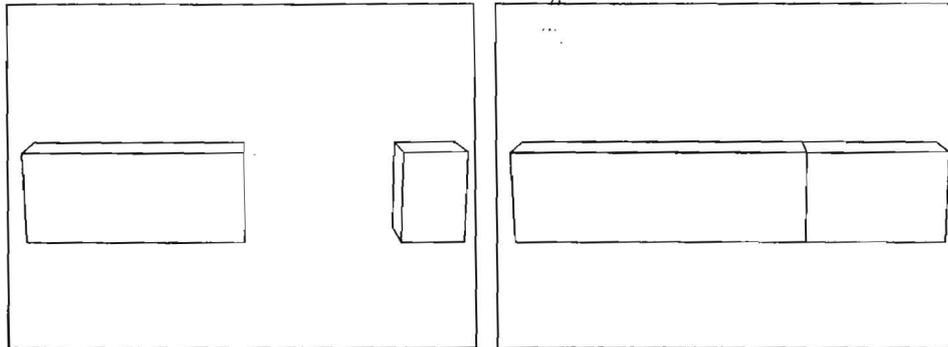


Abb.5.5 (a): Zwei Quader werden abstrahiert,...

Abb.5.5 (b): ...indem beide Objekte um die gleiche Strecke dimensioniert werden ($\lambda = 0.5$).

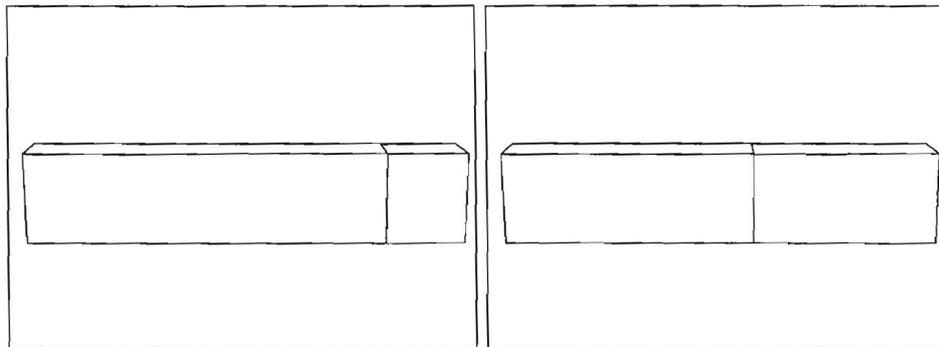


Abb.5.5 (c): ...indem der linke Quader um 90 Prozent „wächst“ ($\lambda = 0.9$).

Abb.5.5 (d): ...indem eine Massengewichtung vorgenommen wird (λ s.o.).

Durch das *Verschmelzen* sollen Formattribute beider Objekte in einem neuen Objekt reflektiert werden. PROXIMA kann dazu auf zwei Methoden zurückgreifen. Sollen beide Objekte gleichwertig im Abstraktionsresultat berücksichtigt werden, so wird das Ergebnis durch die konvexe Hülle der beiden Objekte gebildet. Wenn hingegen eines der beiden Objekte stärker im Ergebnis berücksichtigt werden soll, wird nur in die Richtung verschmolzen, in

der das andere Objekt liegt. Es wird also nur eine Dimension des zweiten Objekts im Ergebnis berücksichtigt. Beide Methoden sind parametrisierbar. Beispiele für verschiedene Verschmelzungen von zwei Zylindern sind in Abbildung 5.6 zu sehen.

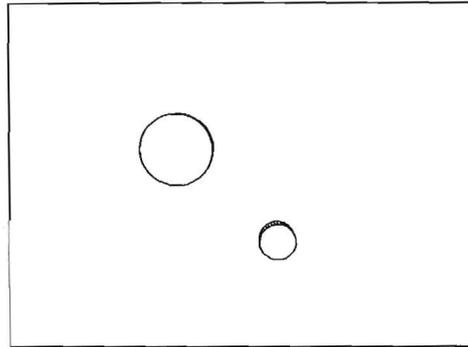


Abb.5.6 (a): Die Ausgangslage.

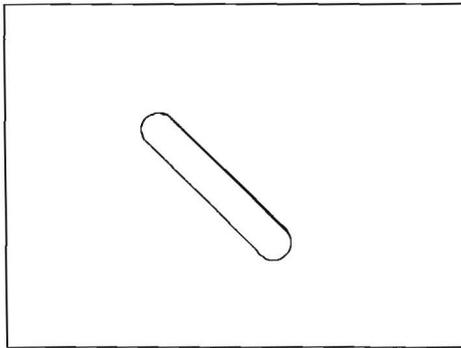


Abb.5.6 (b): Der kleinere Zylinder dominiert im Ergebnis.

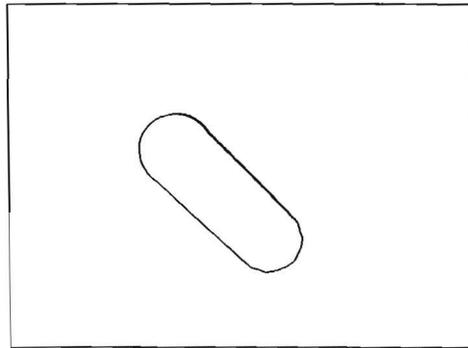


Abb.5.6 (c): Der größere Zylinder dominiert im Ergebnis.

5.4.2.3 Durchführung der Abstraktion

Die technische Realisierung der Abstraktion basiert auf den räumlichen Beziehungen der Objekte und wird durch die Darstellungsbedingungen der betroffenen Objekte beeinflusst. Es werden während der Modellabstraktion nur

jeweils benachbarte Objekte in den Abstraktionsprozeß miteinbezogen. Da die Abstraktionstypen der Modellebene als binäre Operationen realisiert wurden, sind in einen grundlegenden Abstraktionsschritt immer genau zwei Objekte involviert.

Der Einsatz von parametrisierten Abstraktionstypen ermöglicht die Einhaltung der Darstellungsbedingungen. In der jetzigen Version wird die Wahl der Parameterwerte durch Regeln realisiert. Dabei arbeitet PROXIMA mit einem Basisregelpaket, das Regeln enthält, die aufgrund der Darstellungsbedingungen der Objekte die Parameter für die Abstraktionstypen setzen.

In der nachfolgenden Regel werden z.B. die Abstraktionsparameter für zwei Objekte gesetzt. Objekt1 soll dabei wahrnehmbar und Objekt2 diskriminierbar bleiben. Im Aktionsteil der Regel wird für das Dimensionieren von Objekt1 ein Wert für λ von 0.7 gewählt und es wird festgelegt, daß beide Objekte nicht verschmelzen dürfen (dies ergibt sich aus der Tatsache, daß Objekt2 diskriminierbar bleiben soll).

```
(REGEL Objekte-wahrnehmbar-und-diskriminierbar
  (Bedingung
    (Darstellungsbedingung ?Objekt1 'wahrnehmbar)
    (Darstellungsbedingung ?Objekt2 'diskriminierbar)
  )
  (Aktion
    (Abstrahiere-Objekte ?Objekt1 ?Objekt2 0.7 'Keine-Verschmelzung)
  )
)
```

Der große Vorteil des regelbasierten Ansatzes liegt in der Möglichkeit, dieses Regelpaket beliebig um Regeln zu erweitern, die noch zusätzlich Kontextinformation berücksichtigen, um so die Parameterwahl zu perfektionieren.

Natürlich müssen im Abstraktionsprozeß auch geometrische Unterschiede der drei Objektprimitiven (Quader, Zylinder und Lamina) berücksichtigt werden, bzw. ihr Einfluß untereinander während der Abstraktion. Werden

z.B. Quader oder Zylinder (wie man in Abbildung 5.7 sieht.) miteinander verschmolzen, so ist das Abstraktionsergebnis unterschiedlich, auch wenn die Darstellungsbedingungen für alle Objekte gleich sind. Ähnlich wie dies auch im System GRUFF2 (siehe Abschnitt 3.2.4) realisiert wurde, existiert für jedes Grundprimitiv eine eigene Abstraktionsprozedur, die diesen geometrischen Unterschieden Rechnung trägt.

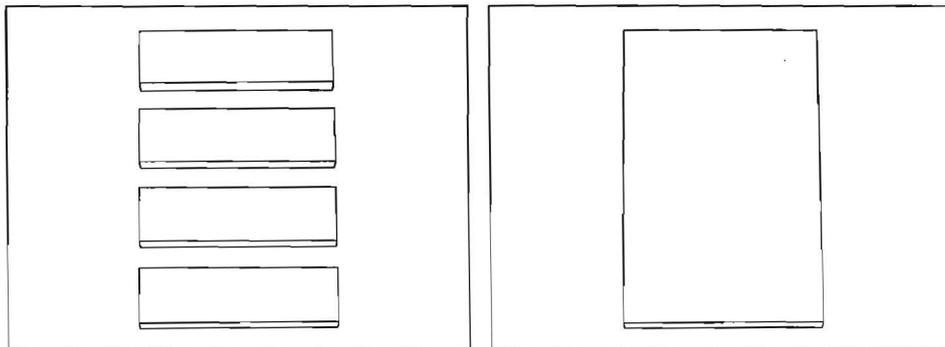


Abb.5.7 (a): Vier Quader werden.. Abb.5.7 (b): verschmolzen.

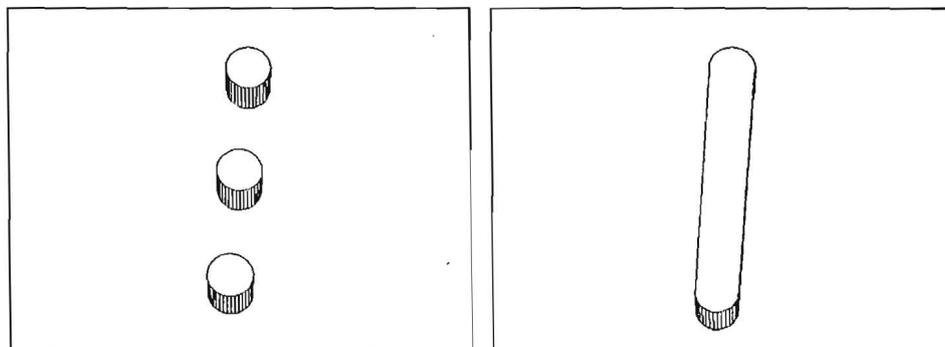


Abb.5.7 (c): Drei Zylinder werden.. Abb.5.7 (d): ... verschmolzen.

Ein wenig komplizierter wird es, wenn zwei unterschiedliche Grundprimitive Gegenstand der Abstraktion sind. Abbildung 5.8 zeigt die Verschmelzung eines Quaders mit einem Zylinder. Es können mehr geometrische Attribute des

Quaders oder des Zylinders im Abstraktionsergebnis erscheinen¹⁰. Diese verschiedenen Möglichkeiten werden von PROXIMA berücksichtigt und können gezielt in Abhängigkeit der Darstellungsbedingungen benutzt werden.

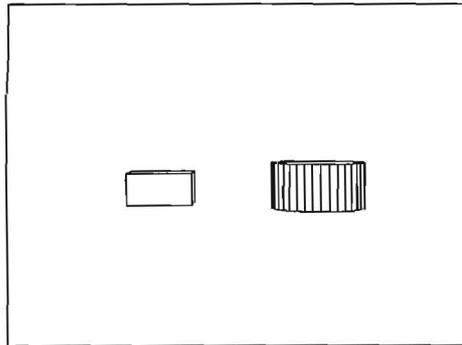


Abb.5.8 (a): Ausgangslage

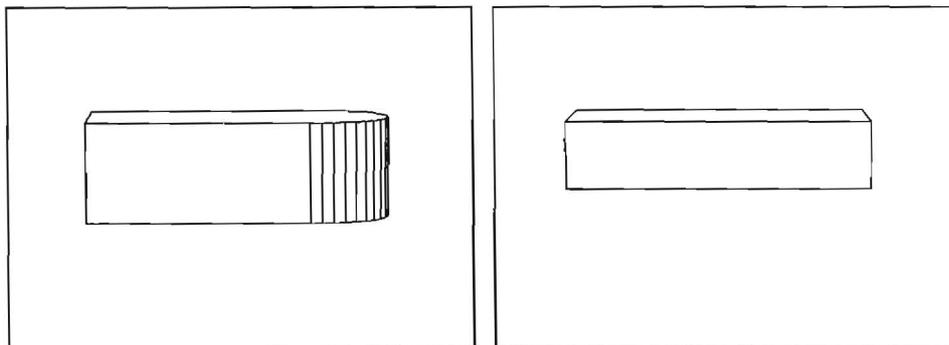


Abb.5.8 (b): Die Attribute des Zylinders dominieren das Abstraktionsergebnis.

Abb.5.8 (c): Die Attribute des Würfels dominieren das Abstraktionsergebnis.

Ein Wort noch zu dem Grundprimitiv Lamina. Laminas haben zunächst kein Volumen, können aber während des Dimensionierens zu Quadern werden. Im weiteren Verlauf werden sie dann ebenso behandelt.

¹⁰Daraus resultieren auch die unterschiedlichen Größen der Objekte in Abbildung 5.8.

Das endgültige Abstraktionsergebnis wird Schritt für Schritt erzeugt. So kann entweder durch eine übergeordnete Komponente, oder durch einen menschlichen Benutzer eine Kontrolle vorgenommen werden. Die gesamte Abstraktion verläuft nach folgenden Grundsätzen:

- Es wird „bottom-up“ in der Zusammenbauhierarchie vorgegangen. Dabei werden zunächst die Grundprimitive der zusammengesetzten Objekte abstrahiert. Wurden alle Grundprimitive eines zusammengesetzten Objekts verschmolzen, so besteht dieses wieder aus einem Grundprimitiv (z.B. aus einem deformierten Zylinder), das nun seinerseits dimensioniert oder verschmolzen werden kann. Zusammengesetzte Objekte, die noch aus mehreren Grundprimitiven bestehen, werden nicht dimensioniert oder verschmolzen.
- In einem Abstraktionsschritt wird ein Objekt entweder dimensioniert, verschmolzen oder weggelassen. Dadurch wird gewährleistet, daß zwischen zwei Schritten der Grad der Abstraktion kontinuierlich wächst.

Nach jedem Abstraktionsschritt müssen die räumlichen Beziehungen neu berechnet werden, und eine Evaluierungsroutine prüft, welche Objekte noch weiter abstrahiert werden dürfen. Mit diesen Objekten wird der nächste Abstraktionsschritt durchgeführt. Dies geschieht solange, bis kein Objekt mehr abstrahiert werden darf; der höchste Abstraktionsgrad ist erreicht, und alle Darstellungsbedingungen sind ausgereizt.

Die Modellabstraktion ist im folgenden Pseudocode zusammengefaßt.

```

PROC Modellabstraktion
LOOP
  MODULE Evaluieren, welche Objekte abstrahiert werden dürfen
  IF Kein Objekt darf abstrahiert werden
    THEN END
    ELSE MODULE Berechne räumliche Beziehungen
      LOOP for Objekte
        MODULE Abstrahiere Objekt mit Nachbarobjekt
        POOL
      FI
    POOL
  FI
POOL
CORP

```

5.4.3 Die Projektionsabstraktion

Die Abstraktion auf der Projektionsebene wird schon während der Bearbeitung des lokalen Kontextes mitbestimmt. Hierbei spielen Vorder- und Hintergrundobjekte sowie die Darstellungsbedingungen eine entscheidende Rolle. Aus diesem Grund kann die Projektionsabstraktion entweder direkt nach der Bestimmung des lokalen Kontextes, oder aber auch erst nach der Modellabstraktion durchgeführt werden. Insbesondere ist es möglich, Modellabstraktionen weiter auf die Projektionsebene zu übertragen, um so das Abstraktionsresultat zu verstärken. Kombinierte Abstraktionen (auf der Modell- und Projektionsebene) liefern meistens gute Ergebnisse. Da aber beide Abstraktionsarten grundsätzlich unabhängig voneinander arbeiten, kann es Fälle geben, in denen verwirrende Resultate erzeugt werden.

5.4.3.1 Parametrisierung der Abstraktionstypen

Während der Projektionsabstraktion finden die Abstraktionstypen *Vereinheitlichen* und *Verändern der Strichstärken* Verwendung. Dazu werden die Kanten von Objekten gestrichelt und/oder ihre Strichstärken verändert. Durch die Veränderung von Parametern können die Strichstärken sowie die Strichabstände gestrichelter Objekte beeinflusst werden. Abbildung 5.9 zeigt unterschiedliche Möglichkeiten, eine IC-Bank auf der Projektionsebene zu abstrahieren.

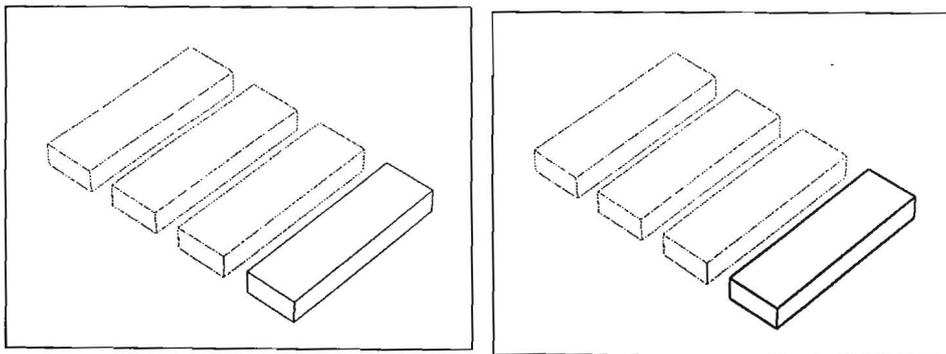


Abb.5.9 (a): Drei der ICs werden gestrichelt, wodurch ein IC in den Vordergrund tritt.

Abb.5.9 (b): Wird die Strichstärke des ICs verändert, dann kann dieser Effekt noch verstärkt werden.

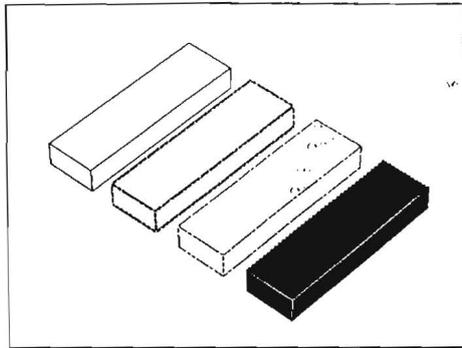


Abb.5.9 (c): Eine Kombination aus Strichstärken und Invertierung ist ebenfalls möglich.

Objekte, die besonders hervorgehoben werden sollen, können invers dargestellt werden. Farben werden derzeit im Abstraktionsprozeß von PROXIMA aus technischen Gründen nicht berücksichtigt, können aber in der Abstraktion durchaus eine Rolle spielen (siehe dazu [20] und Abschnitt 8.2.2).

Der Algorithmus zur Projektionsabstraktion bearbeitet nacheinander jedes Objekt in der Bildszene und führt die Abstraktionen entsprechend der Parameter, wie sie im lokalen Kontext gesetzt wurden, durch. Dies ergibt folgenden Pseudocode:

```

PROC Projektionsabstraktion
  LOOP über alle Objekte der Bildszene
    COND lokaler Kontext
      THEN MODULE Strichel Objekt
      THEN MODULE Invertiere Objekt
      THEN MODULE Verändere Strichstärke von Objekt
    DNOC
  POOL
CORP

```

Kapitel 6

Arbeiten mit PROXIMA an Anwendungsbeispielen

Die vielfältigen Einsatzmöglichkeiten eines Abstraktionssystems spiegeln sich in diesem Kapitel wider. Anhand einer Reihe von Anwendungsbeispielen wird die Funktions- und Ergebnisvielfalt von PROXIMA demonstriert.

6.1 Die Beispieldomäne

Die geometrischen Modelle, mit denen PROXIMA getestet wurde, stammen alle aus dem WIP-Projekt [65], in dessen Rahmen diese Arbeit entstand. Dabei handelt es sich um das Modell eines Rasenmähers, einer Kaffeemaschine und eines Modems. Die Platine des Modems (siehe Abbildung 6.1) erwies sich in diesem Zusammenhang als besonders geeignet, graphische Abstraktionsvorgänge zu verdeutlichen. Im Gegensatz zu den anderen Modellen wurde deshalb für die Modemplatine eine ausführliche Datenbank über signifikante Attribute angelegt. Abstraktionen für die beiden anderen Domänen können daher nur auf der Basis von rein geometrischen Betrachtungen vorgenommen werden.

6.2 Betriebsmodi

Grundsätzlich lassen sich mithilfe von PROXIMA Abstraktionen auf zwei unterschiedliche Arten und Weisen erstellen. Es besteht sowohl die Möglich-

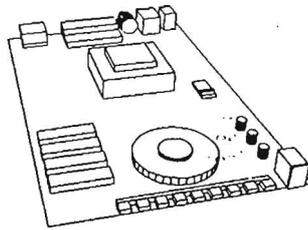


Abbildung 6.1: Das Modell einer Modemplatine als Testdomäne.

keit, diese durch einen interaktiven Dialog mit dem System zu erzeugen, als auch den gesamten Abstraktionsvorgang vollautomatisch durchführen zu lassen. Die daraus resultierenden Betriebsmodi werden im folgenden genauer dargestellt.

6.2.1 Interaktiver Betriebsmodus

Die Ansteuerung von PROXIMA erfolgt über eine graphische Benutzungsschnittstelle, die ausführlicher in Kapitel 7 erläutert wird. Ein Teil dieser Schnittstelle besteht aus einem Graphikausgabefenster, in dem Abstraktionsvorgänge und Resultate visualisiert werden. Der Benutzer hat die Möglichkeit, mit einer Zeigevorrichtung sichtbare Objekte direkt im Ausgabefenster zu selektieren. Die nachfolgende Abbildungssequenz verdeutlicht die interaktive Vorgehensweise.

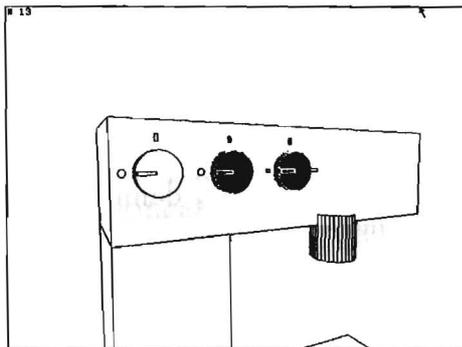


Abb.6.2 (a): Die Knöpfe werden selektiert.

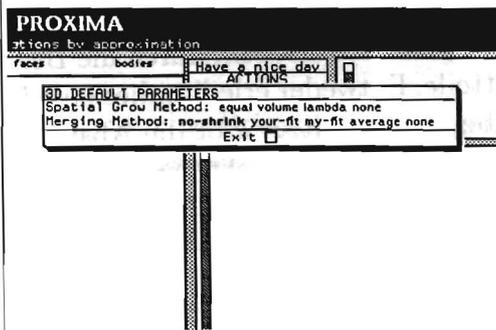


Abb.6.2 (b): Die Abstraktionsparameter werden eingestellt.

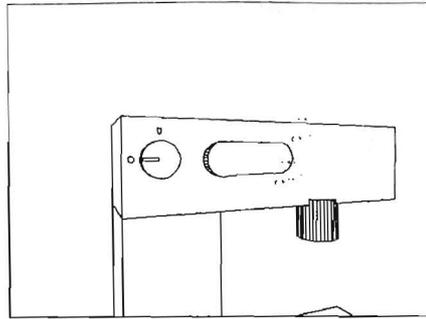


Abb.6.2 (c): Die Abstraktion wird durchgeführt.

Diese Auswahlmethode erlaubt dem Benutzer zu bestimmen, welche Objekte einer Abstraktion unterworfen werden sollen und welche nicht. Darüber hinaus gestattet die Benutzungsschnittstelle die Beeinflussung und Variation einer Vielzahl von Parametern, so daß das Abstraktionsergebnis auf die jeweiligen Wünsche des Benutzers zugeschnitten werden kann.

Signifikante Attribute spielen im interaktiven Betriebsmodus keine Rolle. Allein der Benutzer entscheidet, welche Details in welcher Form abstrahiert werden, und kann so Schritt für Schritt das gewünschte Abstraktionsresultat erzeugen.

6.2.2 Vollautomatischer Betriebsmodus

Sollen Abstraktionen vollautomatisch erzeugt werden, so muß zunächst die Abstraktionsaufgabe an das System formuliert werden. Dies geschieht, wie in Abschnitt 5.3 beschrieben, durch die Beschränkung der Darstellung einzelner Objektteile. Entweder erledigt dies eine übergeordnete Planungskomponente, oder aber der Benutzer gibt die Abstraktionsaufgabe direkt vor. PROXIMA erzeugt, unter Berücksichtigung der signifikanten Attribute, dann ein Abstraktionsresultat, in dem alle Darstellungsbedingungen erfüllt sind.

6.2.3 Halbautomatischer Betriebsmodus

Die schrittweise Erhöhung des Abstraktionsgrades (siehe dazu die Erläuterungen aus Kapitel 5) erlaubt hier eine Mischung beider Betriebsmodi. Der vollautomatische Abstraktionsvorgang kann in einem Zwischenergebnis unterbrochen werden, um dieses dann interaktiv weiterzuverarbeiten. So kann

PROXIMA dem Benutzer Abstraktionsvorschläge unterbreiten, die editiert werden können.

6.3 Einsatzfelder

Im Bereich der Graphikgenerierung stellen Abstraktionen ein wichtiges gestalterisches Mittel dar, um Inhalte gezielt zu visualisieren. Durch die beiden unterschiedlichen Betriebsmodi kann PROXIMA in eine Reihe von Anwendungen integriert werden.

6.3.1 Subkomponente eines Intellimediasystems

Die Komplexität der Information, mit der Benutzer von technischen Geräten umgehen müssen, wächst stetig an. Aus diesem Grund steigt auch der Bedarf an Informationssystemen, die möglichst in verschiedenen Modi (z.B. Sprache, Text und Graphik) diese Information geeignet vermitteln können. Dabei kommt es darauf an, diese Modi in ergänzender Weise einzusetzen und zu koordinieren.

WIP (*Wissensbasierte InformationsPräsentation* [65]) ist ein Vertreter für ein wissensbasiertes Informationssystem, das aus diesen Gründen entwickelt wurde.

Die Information, die von einem Anwendungssystem (z.B. einem Leitstand oder Hilfesystem) zu einem Benutzer fließt, wird in WIP unter der Berücksichtigung von Generierungsparametern (z.B. Zielgruppe oder Zielsprache) so aufbereitet, daß eine Präsentation entsteht, in der mehrere Modi integriert sind.

Zwei Arten von Wissen werden benötigt, um eine Präsentation zu generieren. Zum einen domänenabhängiges Anwendungswissen und zum anderen das von der Eingabe unabhängige Wissen über die Lösung von Präsentationsaufgaben.

WIP besteht aus fünf Hauptkomponenten (siehe Abbildung 6.3):

- einem Präsentationsplaner [4],
- einem Layout-Manager [27],
- je einem Generator für Graphik [50] und Text [34],

- sowie einer Komponente zur Verwaltung und zum Schlußfolgern über repräsentiertes Domänenwissen [28].

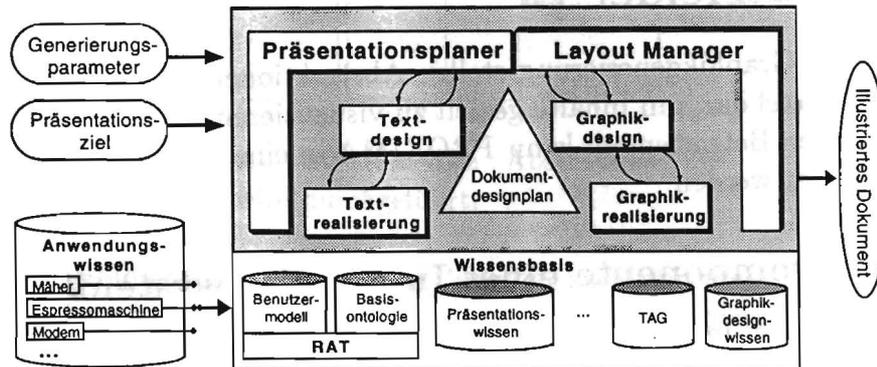


Abbildung 6.3: Die Architektur des WIP-Systems.

Der Präsentationsplaner ist für den Inhalt der Präsentation verantwortlich. Er legt in einem hierarchischen Planungsverfahren fest, wie die zur Verfügung stehenden Modi koordiniert werden sollen und steuert dann die entsprechenden Designkomponenten an (derzeit Text und Graphik). Dies geschieht durch die Formulierung von elementaren Präsentationsaufgaben (z.B. eine verbale Aufforderung formulieren oder ein Objekt graphisch darstellen). Die Designkomponenten der Generatoren erzeugen daraus eine Sequenz von Instruktionen, die von den Realisierungskomponenten ausgeführt werden.

Beim Graphikdesign werden die Aufgaben an die Realisierungskomponente so formuliert, daß Präsentationsaufgaben nicht direkt Bildern zugeordnet werden, sondern diese vielmehr durch die Vorgabe von Anforderungen (*graphische constraints*) spezifiziert werden. So wird die Menge der möglichen Designs immer weiter eingeschränkt, bis nur noch wenige Möglichkeiten in Frage kommen. Jedoch bleibt die Option erhalten, daß mehrere Präsentationsziele gleichzeitig erfüllt werden, wenn alle entsprechenden Anforderungen eingehalten werden. Diese Vorgehensweise unterstützt die inkrementelle Generierung von Präsentationen durch das WIP-System, der folgende Motivation zu Grunde liegt (vgl. dazu auch [4]):

- Der Wunsch nach On-line Präsentationen

- Technische Gründe (Die Modularisierung erfordert Kommunikation zwischen den einzelnen Komponenten. Der Planer benötigt Teilergebnisse, um vernünftig weiter arbeiten zu können.)

Die Graphikrealisierungskomponente von WIP integriert 3D-Techniken und 2D-Techniken¹ aus den Systemen TOPAS (einem System zur Visualisierung von Handlungen [53]) und AnnAII (einem wissensbasierten Annotationssystem [68]). PROXIMA ordnet sich ebenfalls hier ein. Der Graphikdesigner kann entsprechend seiner Philosophie eine Spezifikation der Abstraktion durch Darstellungsbedingungen formulieren. Diese Darstellungsbedingungen werden von PROXIMA im *vollautomatischen* Betriebsmodus in eine graphische Abstraktion umgesetzt. Die inkrementelle Vorgehensweise des Systems unterstützt dabei zusätzlich das Gesamtkonzept von WIP. PROXIMA erhöht den Abstraktionsgrad schrittweise, bis die Darstellungsbedingungen ausgereizt sind. In jedem Abstraktionsschritt bleiben also alle Darstellungsbedingungen erfüllt. Sollte der Designprozeß zeitkritisch sein, so kann der Graphikdesigner PROXIMA nach einem beliebigen Abstraktionsschritt unterbrechen.

Die Arbeitsweise im vollautomatischen Betriebsmodus wird nun durch ein kleines Beispiel illustriert (siehe Abbildung 6.4). Nehmen wir an, es wird eine Illustration gewünscht, in der durch Abstraktion die Aufmerksamkeit des Betrachters auf eine bestimmte LED und einen bestimmten Schalter gelenkt wird. Diese Anforderung wird durch folgende Spezifikationen beschrieben:

1. SCHALTER muß *identifizierbar* bleiben.
2. LED muß *identifizierbar* bleiben.
3. Vordergrundobjekte sind SCHALTER und LED.

Erreicht wird dies durch die Abstraktion der Hintergrundobjekte. Um aber einen nicht zu hohen Abstraktionsgrad zu erhalten, werden den Darstellungen von Lautsprecher, LED-Reihe und IC-Bank zusätzliche Restriktionen auferlegt:

4. LAUTSPRECHER muß *sichtbar* bleiben.

¹In einer weiteren Ausbaustufe sollen auch die Funktionen des Animationssystems BETTY [12] ausgenutzt werden.

5. IC-BANK soll *klassifizierbar* bleiben.

6. LED-REIHE soll *klassifizierbar* bleiben.

Das System beginnt mit der Abstraktion auf der Modellebene. Abbildung 6.4(b) zeigt einen ersten Zwischenschritt. Die IC-Bank wurde durch Schließen der Lücken zwischen den einzelnen IC Blöcken (Abstraktionstyp: *Dimensionsnieren*) auf den maximal möglichen Abstraktionsgrad gebracht. Eine weitere Abstraktion der IC-Bank würde die Darstellungsbedingung Nr. 5 (*klassifizierbar*) verletzen. Aus demselben Grund darf die LED-Reihe keiner weiteren Abstraktion unterzogen werden.

Im Verlauf der weiteren Abstraktion „wachsen“ die hinteren Bauteile bzw. die Kondensatorbank und der Lautsprecher aufeinander zu (Abbildung 6.4(c)). Schließlich entstehen durch den Abstraktionstyp *Verschmelzen* aus diesen Bauteilen zwei neue Objekte, und der Schalter vorne links wird eliminiert (Abbildung 6.4(d) dokumentiert diesen Vorgang). Damit ist die Abstraktion auf der Modellebene innerhalb der Darstellungsbedingungen abgeschlossen. Die anschließende Abstraktion auf der Projektionsebene ist in Abbildung 6.4(e) zu sehen. Dabei werden die Hintergrundobjekte gestrichelt und die Vordergrundobjekte mit einer dickeren Strichstärke gezeichnet.

Die abstrahierte Darstellung des Ausgangsmodells erfüllt im Endresultat alle geforderten Darstellungsbedingungen.

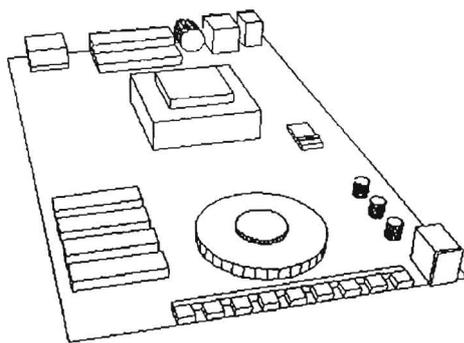


Abb.6.4 (a): Ausgangssituation

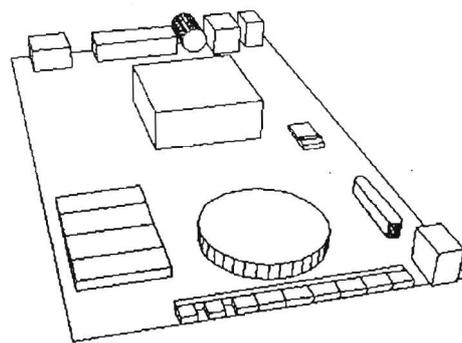


Abb.6.4 (b): 1. Abstraktionsschritt

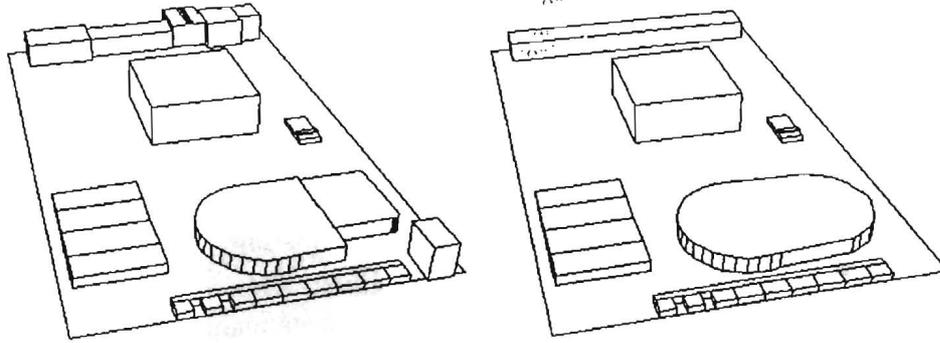


Abb.6.4 (c): 2. Abstraktionsschritt Abb.6.4 (d): 3. Abstraktionsschritt

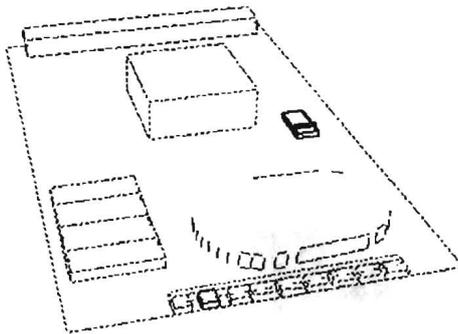


Abb.6.4 (e): Abstraktion auf der Projektionsebene und Endresultat.

6.3.2 Interaktive Generierung von 3D-Illustrationen

Während WIP Graphiken für Bedienungsanleitungen und Hilfesysteme vollautomatisch erzeugt, besteht auch ein Bedarf an Werkzeugen, die solche Bilder in Interaktion mit dem Benutzer (in diesem Fall: einem Graphiker) entstehen lassen. Dabei kann der Grad der Unterstützung, die ein solches System bietet, unterschiedlich sein, wie zwei im folgenden vorgestellten Systeme unterstreichen werden. Beide Systeme basieren auf dem Konzept der WIP Graphikrealisierung und bieten als eine mögliche Illustrationstechnik

die Abstraktion des PROXIMA Systems an.

Im Gegensatz zum Einsatz im WIP Projekt wird in diesen Fällen vom *interaktiven* Betriebsmodus Gebrauch gemacht.

6.3.2.1 Das System TOPAS

In [54] wurde TOPAS um eine umfangreiche Abstraktionskomponente erweitert. Dabei wurde der gesamte Funktionsumfang von PROXIMA unter der graphischen Oberfläche von TOPAS zur Verfügung gestellt. Ziel war es, ein Werkzeug zu entwickeln, das die Erzeugung von Illustrationen für technische Betriebs- und Bedienungsanleitungen erleichtert. Ausgangsbasis für das System sind wiederum Modelldaten, die direkt einem CAD-System entnommen werden können.

Es stehen verschiedene Illustrationstechniken zur Verfügung, die außer der Abstraktion die Anfertigung von beispielsweise Explosionszeichnungen, Aufrißzeichnungen, Pfeilen innerhalb der Graphik und virtuellen Böden erleichtern. Durch Kombination von Techniken lassen sich, wie in Abbildung 6.5 zu sehen, anspruchsvolle Graphiken erzeugen.

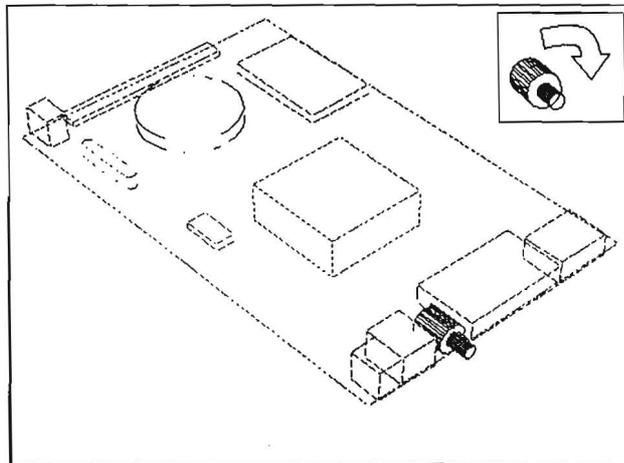


Abbildung 6.5: Eine Kombination der Illustrationstechniken Abstraktion, Pfeil und Inset.

Die modifizierten Modelle dienen auch als Input für weitere Verfahren (z.B. einen Renderer), so daß die resultierenden Illustrationen weiter an Qualität gewinnen. Die folgende Sequenz ist ein Beispiel für gerenderte Abstraktionen.

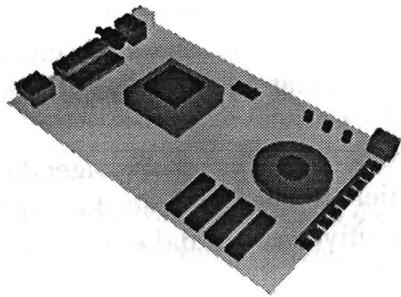


Abb.6.6 (a): Ausgangssituation.

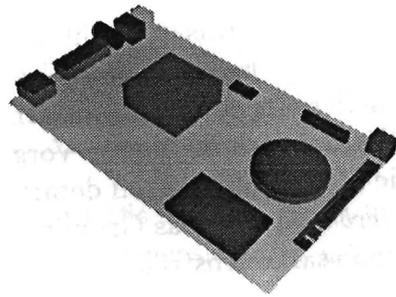


Abb.6.6 (b): 1. Abstraktionsschritt

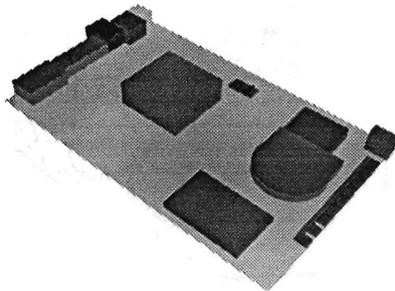


Abb.6.6 (c): 2. Abstraktionsschritt

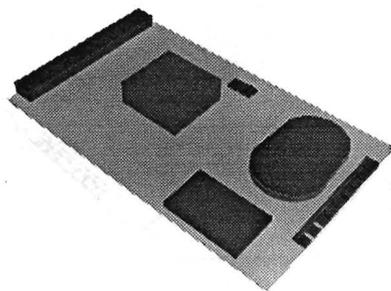


Abb.6.6 (d): 3. Abstraktionsschritt

6.3.2.2 Das System AWI

AWI (**A** Workbench for Semi-Automated Illustration Design) nennt sich ein weiteres System, in dem die Abstraktionskomponenten von PROXIMA Verwendung finden. Auch AWI dient der Realisierung von komplexen Graphiken für technische Bedienungsanleitungen. Es basiert genauso wie TOPAS

auf der Graphikrealisierungskomponente des WIP Projekts. Allerdings versucht AWI einen Mittelweg zu gehen, der den Automatismus der Graphikgenerierung betrifft. Während in WIP Graphiken vollautomatisch entworfen werden und in TOPAS eher durch den Benutzer, so wird in AWI ein halbautomatischer Entwurf unterstützt. Das System macht dem Benutzer einen Vorschlag, den dieser editieren kann und dann auf Wunsch von dem System noch evaluieren lassen kann. Diese Vorgehensweise berücksichtigt, daß ein vollautomatischer Ansatz nicht in der Lage ist, alle Nebenbedingungen (wie ästhetische Faktoren) bei der Graphikgenerierung zu berücksichtigen, und ein manueller Ansatz oft zu unkomfortabel ist.

Zur Verdeutlichung dieser Vorgehensweise betrachte man folgendes Beispiel: Die Modemplatine soll derart abstrahiert werden, daß der Lautsprecher *identifizierbar* bleibt. Das Grundabstraktionsniveau wird dabei *niedrig* gehalten, um zu starke Abstraktionen zu verhindern.

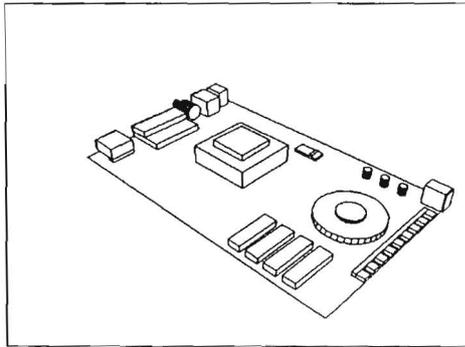


Abb.6.7 (a): Die Ausgangssituation der Abstraktion.

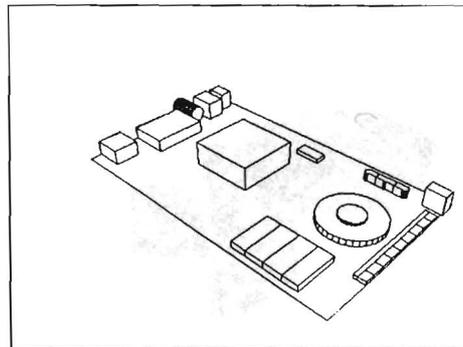


Abb.6.7 (b): Die IC- und Kondensatorbank sowie die LED-Reihe werden durch Dimensionieren abstrahiert. Der Transformator verschmilzt zu einem Objekt.

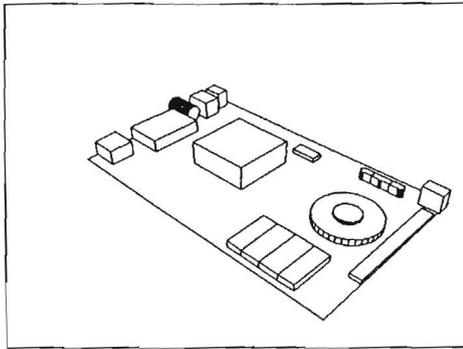


Abb.6.7 (c): Die LED-Reihe und die hinteren Bauteile verschmelzen zu einzelnen Einheiten.

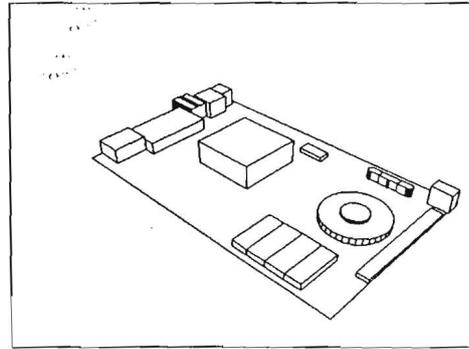


Abb.6.7 (d): Die hinteren Bauteile werden durch Dimensionieren weiter abstrahiert.

Abbildung 6.7(a) zeigt das ursprüngliche Modell. Die Abbildungen 6.7(b) - 6.7(e) dokumentieren die schrittweise Arbeitsweise von PROXIMA. In diesem Beispiel genügt die Stärke der Abstraktion dem Benutzer noch nicht. Er selektiert die IC- und Kondensatorbank (Abbildung 6.7(f)) und führt einen weiteren Abstraktionsschritt aus, dessen Ergebnis in Abbildung 6.7(g) und 6.7(h) zu sehen ist.

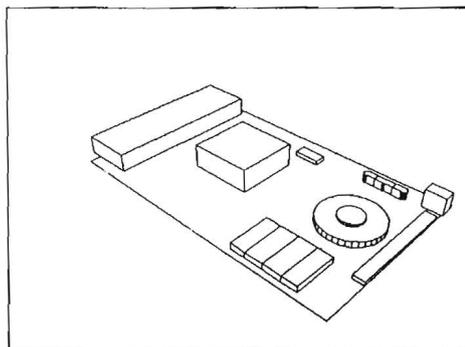


Abb.6.7 (e): Das letzte Bild des Vorschlags: die hinteren Bauteile verschmelzen und der vordere Schalter wird eliminiert.

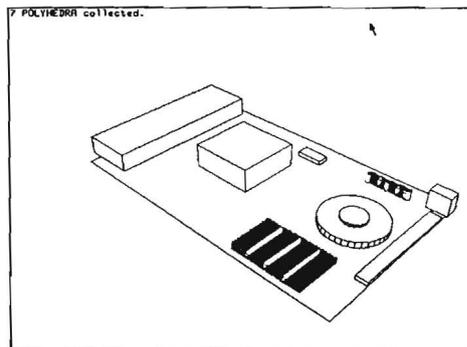


Abb.6.7 (f): Der Benutzer selektiert die IC- und Kondensatorbank.

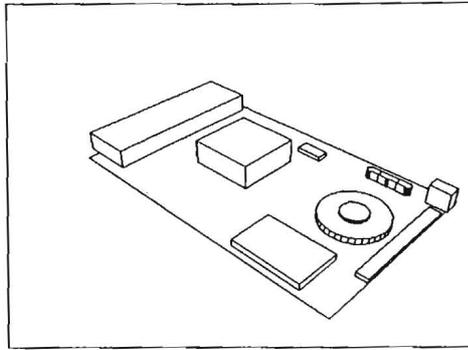


Abb.6.7 (g): Die IC-Bank wird weiter abstrahiert.

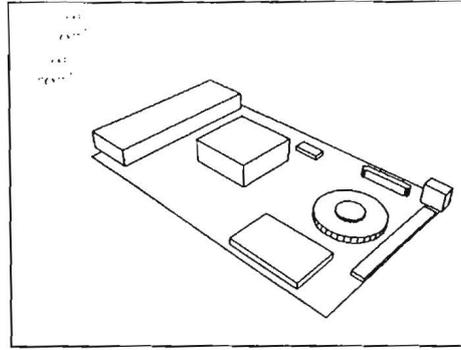


Abb.6.7 (h): Die Kondensatorbank wird weiter abstrahiert.

6.3.3 Animationen

Der Einsatz von Abstraktionen kann auch technisch motiviert sein, wie in Abschnitt 2.2 erläutert wurde. Ein Beispiel ist der Einsatz von Abstraktionen im Laufe der Generierung von Animationen.

3D-Animationssysteme (wie z.B. BETTY [12]) arbeiten ebenfalls mit Modelldaten. Dazu werden die entsprechenden Objekte, bzw. die Kamera im Objektraum bewegt und die einzelnen Frames² werden daraus berechnet. Je mehr Objekte bewegt werden müssen oder sich im Bildausschnitt befinden, desto mehr Rechenzeit muß für die Berechnung eines Frames zur Verfügung gestellt werden. Dieser Kostenpunkt vervielfacht sich, wenn die Bilderstellung durch ein photorealistisches Verfahren (z. B. Render oder Radiosity Verfahren) durchgeführt wird.

Die Senkung der Komplexität der Objekte führt auch zu einer Verringerung der Animationsgenerierungsdauer. Abstraktionen können unter zwei Gesichtspunkten eingesetzt werden:

- Berücksichtigen der Bildaussage:
Details, die für die Bildaussage nicht relevant sind, können abstrahiert werden. Diese Vorgehensweise ähnelt sehr der Vorgehensweise bei der Erzeugung von abstrahierten statischen Graphiken und kann daher problemlos durch PROXIMA realisiert werden.

²Einzelbilder der Animation

- Berücksichtigung menschlicher visueller Wahrnehmung:
Details, die vom menschlichen Wahrnehmungsapparat nicht mehr wahrgenommen werden können abstrahiert werden, ohne den Gesamteindruck zu verändern. Bei schnellen Bewegungen werden z.B. kleinere Objekte nicht mehr gesehen. Auch diese Form der Abstraktion kann durch Techniken, die PROXIMA zur Verfügung stellt, realisiert werden. Problematisch bleibt allerdings die Rolle der signifikanten Attribute. Sie beruhen schließlich auf der vollen visuellen Wahrnehmungsfähigkeit eines Menschen. Bei schnellen Animationssequenzen kann sich diese Gewichtung verschieben.

6.3.4 Zusammenarbeit mit einem begrifflichen Referenzierungssystem

Die Zusammenhänge zwischen graphischen und begrifflichen Abstraktionen, die u.a. auch in Abschnitt 3.1.1 deutlich wurden, ermutigen zur Zusammenarbeit mit einem System, das in der Lage ist, natürlichsprachliche Referenzausdrücke zu generieren. Das System GORDIAN [55] erzeugt solche Referenzausdrücke zu graphischen Bildelementen. Die Zusammenarbeit mit PROXIMA ist insbesondere unter zwei Gesichtspunkten besonders interessant:

- PROXIMA stellt GORDIAN für seine *abstrakt* gewählten Begriffe eine graphische Realisierung zur Verfügung.
- GORDIAN erzeugt Referenzausdrücke zu graphischen Abstraktionen, die von PROXIMA generiert wurden.

Besonders der zweite Punkt ist sehr vielversprechend. Denn für viele graphische Abstraktionen existieren keine Begriffe. Mithilfe von GORDIAN könnten diese Begriffe oder wenigstens eine Umschreibung der Abstraktionen gefunden werden. Betrachtet man z.B. den Ausschnitt der Platine in Abbildung 6.8(a), so kann durch eine Referenzierung auf die vier Quader durch den Begriff *ICs*, problemlos ein Bezug zur realen Welt hergestellt werden. Dieser Bezug ist in Abbildung 6.8(b) nicht mehr ohne weiteres nachvollziehbar, da die ICs und andere Teile der Platine abstrahiert wurden. GORDIAN ist nun in der Lage eine Referenzierung auf der graphischen Ebene vorzunehmen, indem er die abstrahierten ICs z.B. als *Quader in der linken unteren Bildecke* bezeichnet.

GORDIAN stellt einen textuellen Kontext zu graphischen Abstraktionen zur Verfügung und ermöglicht so eine variabelere Auswahl der Abstraktionsgrade.

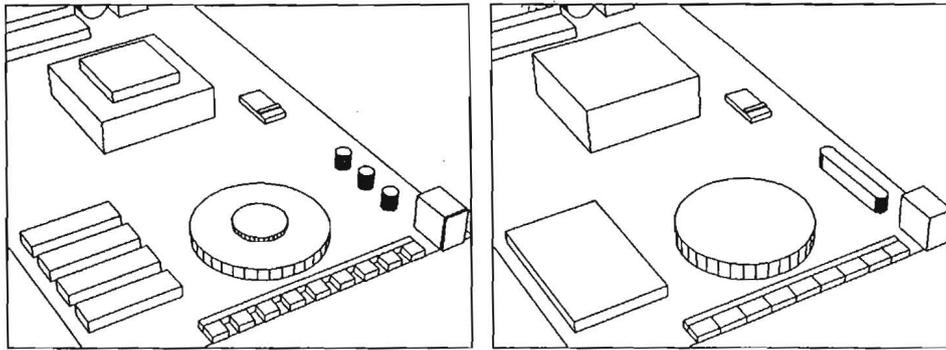


Abb.6.8 (a): Die vier *Ics*

Abb.6.8 (b): Der *Quader* in der linken unteren Bildecke

6.3.5 Graphische Suchsysteme

Die Suche nach graphischem Material in einer Bild- bzw. Graphikdatenbank ist meist ein mühsames Unterfangen. In großen Bildarchiven sind die Bilder meist unter *Schlüsseln* abgelegt [13], um ein späteres Auffinden einer Graphik zu ermöglichen. Dieses Verfahren besitzt einen entscheidenden Nachteil: Die Indizierung muß sehr umfassend sein, denn sonst kann es passieren, daß der Benutzer des Suchsystems, obwohl er noch eine visuelle Vorstellung des Bildes besitzt, den Bildinhalt nicht mehr gemäß der Indizierung beschreiben kann und die Bildsuche fehlschlägt.

In [29] wird ein graphisches Suchsystem vorgestellt, das für zweidimensionale Graphiken, die in einem Zeichnungsbeschreibungsformat vorliegen (in diesem Fall das EPIC-Format) einen Lösungsvorschlag macht. Die Anfrage an das Suchsystem wird dabei nicht textuell, sondern graphisch realisiert. Eine Skizze wird benutzt, um einen Graphikbestand zu durchsuchen. Da diese Skizze aus der Erinnerung des Benutzers gemalt wurde, kann sie naturgemäß nicht alle Details der gesuchten Graphik beinhalten. Ein genaues „Matching“ der Elemente würde aus diesem Grund nicht zum Erfolg führen. Deshalb müssen Details der Graphiken abstrahiert werden, um einen Vergleich durchführen zu können. Abstrahiert wird durch die Vergrößerung der

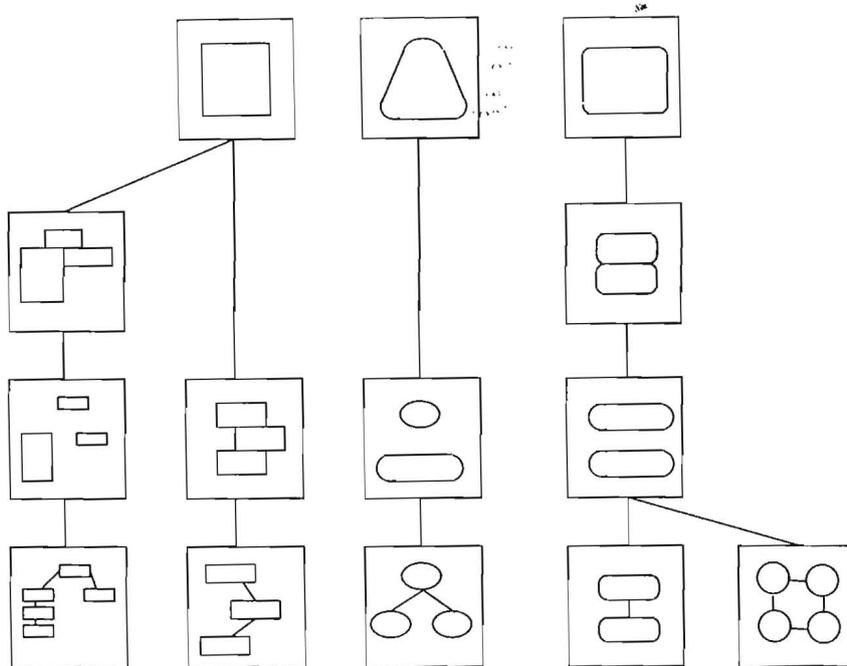


Abbildung 6.9: Über einem Bestand von Graphiken kann mithilfe der Abstraktion eine Suchhierarchie erzeugt werden.

präzisen Werte, die in dem Zeichnungsbeschreibungsformat verwendet werden. Z.B. wird die Größe eines Objekts nicht durch ein Zahlentupel beschrieben, sondern durch eines der Merkmale klein, mittel oder groß. Die Suche wird dabei durch ein assoziatives neuronales Netz realisiert. Daraus ergibt sich eine Reihe von Graphiken, die Spezialisierungen der Eingabe darstellen. Aus diesen Graphiken kann der Benutzer dann die richtige auswählen.

Ein ähnliches Verfahren kann mit graphischer Abstraktion realisiert werden. Dazu erzeugt PROXIMA verschieden starke Abstraktionen der Graphik. Abbildung 6.9 illustriert, wie diese Vorgehensweise über allen Elementen eine hierarchische Struktur definiert. Die Suchanfrage in Form einer Skizze wird dann zunächst mit den Abstraktionen verglichen. Dieses Verfahren verringert die Komplexität der Suche erheblich.

6.3.6 Konvertierung von CAD-Modellen in FEM-Modelle

In Abschnitt 3.2.3 wurde die Notwendigkeit erläutert, während des Konstruktionsverlaufs von Werkstücken detaillierte voluminöse CAD-Modelle durch einen Abstraktionsvorgang in FEM-Modelle zu überführen. Prinzipiell kann diese Konvertierung auch von PROXIMA übernommen werden. Im interaktiven Betriebsmodus können die gewünschten Vereinfachungen mithilfe der Abstraktionstypen *Dimensionieren* und *Weglassen* realisiert werden.

Kapitel 7

Die Implementation von PROXIMA

Das System wurde auf einer Symbolics XL1200 Lispmaschine [61] unter Verwendung des Betriebssystems Genera 8.0.1 vollständig in Common Lisp [58] entwickelt. Es wurde ein objektorientierter Ansatz gewählt, der mithilfe des Symbolics Flavor Systems [60] realisiert wurde.

Die folgenden Abschnitte beschäftigen sich mit Implementationsdetails der einzelnen Komponenten, die in Kapitel 5 vorgestellt wurden.

7.1 Repräsentation der geometrischen Modelle

Die geometrischen Modelle, mit denen PROXIMA arbeitet, wurden mit dem 3D-Editor S-Geometry [62] erstellt. S-Geometry folgt der Kamerametapher und bietet die Möglichkeit, aus verschiedenen Objektprimitiven (Quadern, Zylindern, Laminas, Kugeln und Rotationskörpern) komplexe, zusammengesetzte Objekte zu generieren. Eine zusätzliche Komponente (S-Render) generiert von den Modellen gerenderte Bilder.

S-Geometry ist komplett in Symbolics Common Lisp geschrieben und ermöglicht so einen direkten Zugriff auf alle verwendeten Objektstrukturen. Diese Strukturen enthalten Information über die Flächen, die Kanten und die Knoten der Objekte. Diese sind in einem Flächen-, Kanten- bzw. Knotenring gespeichert. Darüber hinaus stehen viele geometrische Eigenschaf-

ten, wie Volumen, Flächenumfang, Schwerpunkt und Normalenvektor unmittelbar zur Verfügung. Hierdurch wird es möglich, Flächen hinzuzufügen oder zu entfernen. Auch das Verschieben oder das Skalieren von Flächen kann durch die Manipulation der Datenstrukturen erreicht werden. Diese Möglichkeiten werden von PROXIMA während des Abstraktionsvorgangs ausgeschöpft. Z.B. werden bei der Abstraktion eines Zylinders Flächen verschoben und sogar gedreht, um den gewünschten Abstraktionseffekt zu erzielen. Um die Unabhängigkeit vom verwendeten Graphikpaket zu gewährleisten, wurde eine Graphikschnittstelle benutzt. Eine zu enge Verzahnung von PROXIMA mit S-Geometry wurde damit vermieden und die Portierbarkeit für andere Graphiksysteme bleibt erhalten. Eine Umsetzung der Graphikschnittstelle auf eine *Indigo*² von Silicon Graphics wird zur Zeit erwogen. Diese Hardwarekonfiguration mit integriertem Z-Buffer beschleunigt die Darstellung von 3D-Objekten erheblich. Insbesondere die Erzeugung photorealistischer Abbildungen (durch Render- bzw. Radiosity-Verfahren, siehe [24]) werden dadurch stark vereinfacht. Auf dieser Plattform steht auch das 3D-Visualisierungswerkzeug GeoDisplay [45] mit einem erweiterten Funktionsumfang zur Verfügung.

7.1.1 Objektrepräsentationen

PROXIMA verwendet für alle geometrischen Grundprimitive spezielle Datenstrukturen. Alle verwendeten geometrischen Objekte sind sogenannte *Abstract Objects*, die durch die nachfolgende Struktur definiert werden.

```

1(defflavor abstract-object
2      ((sgeo-object nil)
3       (father nil)
4       (sons nil)
5       (3d-constraints (make-instance '3d-constraints))
6       (2d-constraints (make-instance '2d-constraints))
7       )
8      ())
9  :initable-instance-variables
10 :writable-instance-variables)

```

In Zeile 2 wird die Verbindung zwischen abstrahiertem Objekt und dem S-Geometry Objekt hergestellt. Zeile 3 und 4 dienen dem Aufbau der Ab-

straktionshierarchie, die während der Abstraktion erzeugt wird. Zeile 5 und 6 enthalten die Beschränkungen für die Modellabstraktion (3d-constraints) und die Projektionsabstraktion (2d-constraints). Die folgenden Definitionen spezialisieren obige Struktur.

```

1 (defflavor abstract-quader
2   ((face1 (make-instance 'spatial-information))
3     (face2 (make-instance 'spatial-information))
4     (face3 (make-instance 'spatial-information))
5     (face4 (make-instance 'spatial-information))
6     (face5 (make-instance 'spatial-information))
7     (face6 (make-instance 'spatial-information))))
8   (abstract-object)
9   :initable-instance-variables
10  :writable-instance-variables)

```

Das Objekt *Abstract Quader* enthält Information über die räumlichen Relationen seiner 6 Flächen (Zeile 2-7).

```

1 (defflavor abstract-cylinder
2   ((face1 (make-instance 'spatial-information))
3     (face10 (make-instance 'spatial-information))
4     (face19 (make-instance 'spatial-information))
5     (face28 (make-instance 'spatial-information))
6     (face37 (make-instance 'spatial-information))
7     (face38 (make-instance 'spatial-information))))
8   (abstract-object)
9   :initable-instance-variables
10  :writable-instance-variables)

```

Bei der Definition des *Abstract Cylinders* ist gut zu erkennen, welche Referenzflächen benutzt werden (Zeile 2-7).

```

1 (defflavor abstract-lamina
2   ((face1 (make-instance 'spatial-information))
3     (face2 (make-instance 'spatial-information))))
4   (abstract-object)
6   :initable-instance-variables
7   :writable-instance-variables)

```

Für die *Abstract Lamina* gilt das gleiche, wie für die anderen Objekte. Sie besitzt allerdings nur zwei Flächen (Zeile 2 und 3).

Nachzutragen bleiben noch die Definitionen für die räumlichen Relationen und die Beschränkungen:

```
1 (defflavor spatial-information
2   ((nearest-abstract-object nil)
3     (facenr nil)
4     (distance nil)
5     (orientation nil)
6   )
7   ())
8 :initable-instance-variables
9 :writable-instance-variables)
```

Zeile 2 enthält das benachbarte Objekt. In Zeile 3 wird die Nummer der Fläche gespeichert, die dem Objekt zugewandt ist. Geometrische Sachverhalte, wie Entfernung (Zeile 4) und der Normalenvektor der Fläche (Zeile 5) werden ebenfalls berücksichtigt.

```
1 (defflavor 3d-constraints
2   ((visual-constraint nil)
3     (spatial-move nil)
4     (merge nil))
5   ())
6 :initable-instance-variables
7 :writable-instance-variables)
```

Die Beschränkungen für die Modellabstraktion beinhalten die Darstellungsbedingung (Zeile 2), sowie die Parametrisierung des Abstraktionstyps *Dimensionieren* (Zeile 3) und des Abstraktionstyps *Verschmelzen* (Zeile 4).

```
1 (defflavor 2d-constraints
2   ((thickness 0)
3     (dashvector '(1 1))
4   )
5   ())
6 :initable-instance-variables
7 :writable-instance-variables)
```

Für die Abstraktion auf der Projektionsebene sind die Strichstärken (Zeile 2) und die Beschaffenheit der Strichelung (Zeile 3) von Bedeutung.

7.1.2 Datenstrukturen für signifikanten Attribute

Die Implementation der Datenstrukturen, in denen die signifikanten Attribute der Domäne gespeichert werden, wurde durch Assertionen realisiert. Zugriffe werden durch Symbolvergleich durchgeführt.

Es folgt ein Beispiel:

```
1 (cs::add-new-assertion
2 '(ic-bank-1 ((ic-1 ic-2 ic-3 ic-4) () () () ()) ;; identifizierbar
3             (( ) (ic-1 ic-2 ic-3 ic-4) () ()) ;; klassifizierbar
4             (( ) () (ic-1 ic-2 ic-3 ic-4)()) ;; diskriminierbar
5             (( ) () (ic-1 ic-2 ic-3 ic-4)()) ;; wahrnehmbar
6             (( ) () () (ic-1))) *attribute-db*);; sichtbar
```

Dies ist der Datenbankeintrag für die IC-Bank der Modemplatine. Zeile 1 enthält die Darstellungsbeschränkungen für die Teilobjekte, falls die IC-Bank identifizierbar bleiben soll. Dies geschieht durch Listen. In diesem Fall stehen alle Teilobjekte (IC1-IC4) in der ersten Teilliste und erhalten die Darstellungsbeschränkung *identifizierbar*. Die übrigen Teillisten stehen für die vier nachfolgenden Darstellungsbedingungen und sind leer. Wird die Darstellung der IC-Bank anders beschränkt, muß der entsprechende Eintrag (Zeile 3-6) berücksichtigt werden. Damit z.B. die IC-Bank sichtbar bleibt, genügt es, daß IC1 sichtbar bleibt (Zeile 6).

7.2 Regelinterpreter

Sowohl die Erstellung des lokalen Kontexts, als auch die Parametrisierung der Abstraktionstypen stützen sich auf einen regelbasierten Prozeß. PROXIMA greift auf einen Regelinterpreter zurück, der mehrfach innerhalb der Graphikrealisierungskomponente von WIP verwendet wird¹. Dieser Regelinterpreter basiert auf schon existierenden Ansätzen. Die Konfliktauflösungsstrategie lehnt sich an den MEA-Algorithmus von OPS5 [14] an und verwendet

¹Z.B. von den Subsystemen AnnAII und TOPAS.

das Paradigma der offenen Syntax von YAPS [2].

Ein wichtiges Merkmal ist die Fähigkeit des Regelinterpreters, abgeleitete Lösungen zu bewerten. Dies geschieht, ähnlich wie bei XPS-Shell S.1 [63], durch *certainty factors*.

Das Regelpaket zur Propagierung des lokalen Kontexts umfaßt zur Zeit 22 Regeln, und zum Setzen der Abstraktionsparameter wurden 38 Produktionen implementiert.

7.3 Abstraktionsalgorithmen

Dieser Abschnitt behandelt Implementationsdetails der Abstraktionsvorgänge. Es wird dabei davon ausgegangen, daß der lokale Kontext bereits ermittelt wurde und die räumlichen Beziehungen der Objekte durch geometrische Betrachtungen bekannt sind.

7.3.1 Modellabstraktion

Der objektorientierte Ansatz, der bei der Implementation des gesamten Systems verwendet wurde, bietet gerade bei der Implementation der Abstraktionsvorgänge auf der Modellebene große Vorteile. Die geometrischen Unterschiede der Grundprimitive lassen sich mithilfe von Methoden einfach abbilden. So existiert für die Abstraktionstypen *Dimensionieren* und *Verschmelzen* jeweils eine Methode, die den Abstraktionsvorgang durchführt und dabei die geometrischen Besonderheiten der jeweiligen Objekte berücksichtigt.

Eine Hauptabstraktionsmethode für jedes Primitiv steuert den Einsatz der Abstraktionstechniken. Der große Vorteil dieses Ansatzes ist, daß neue Grundprimitive (z. B. Polyhedrons) relativ einfach in den Abstraktionsprozeß mitaufgenommen werden können. Es muß nur die entsprechende Version der Abstraktionsmethode für das neue Grundprimitiv implementiert werden, so daß seine geometrischen Besonderheiten Berücksichtigung finden.

Da die Abstraktionsoperationen binär sind, sich aber eine bestimmte Methode nur auf ein Primitiv beziehen kann, wird die binäre Operation durch einen rekursiven Aufruf realisiert. So kann jedes Grundprimitiv mit jedem anderen interagieren (z.B.: verschmelzen). Stellvertretend wird nun die Abstraktionsmethode für den Quader vorgestellt:

```

1 (defmethod (3d-abstract-object! abstract-quader)
2     (&key
3         (root nil)
4         (exception-list nil)
5         (respond nil)
6         (merge-which nil)
7         (merge-face nil)
8         (merging-type 'no-shrink)
9         (moving-type 'equal)
10        (my-lambda 0.5)
11    )

```

Root ist das übergeordnete Objekt, dem der Quader angehört. Die *exceptionlist* enthält Objekte, die in den Abstraktionsprozeß nicht miteinbezogen werden sollen. Das Prädikat *respond* wird gesetzt, um einen rekursiven Aufruf der Methode zu kennzeichnen. In den Zeilen 6-10 werden Abstraktionsparameter gesetzt.

```

12 (Let ((abstract-copy )
13     (nearest-object)
14     (distance)
15     (orientation)
16     (help-list (make-face-list))
17 (if (not respond)
18     (setf help-list (sort-faces-for-distance (help-list))))
19 (loop for face in help-list
20     if (test-if-face-abstraction-possible face)
21     do (progn (set-context *ri* *db*
22                 '(geometric
23                   ,self
24                   ,(nearest-abstract-object face))))
25         (set-abstraction-parameters-from-db self)
26         (set-face-parameters face)

```

In Zeile 12-16 werden geometrische Hilfsvariablen eingeführt. In den Zeilen 17 und 18 wird im Falle eines nichtrekursiven Aufrufs, eine günstige Abarbeitungsreihenfolge der Seiten festgelegt. Ein rekursiver Aufruf impliziert

diese Reihenfolge schon. Ab Zeile 19 beginnt eine Schleife über alle Seiten des Quaders. In Zeile 20 wird getestet, ob eine Abstraktion überhaupt geometrisch für die Fläche möglich ist². In den Zeilen 21-24 wird der Regelinterpreter aktiviert (Funktionsaufruf: `set-context`), um die Abstraktionsparameter zu ermitteln. Diese werden dann in Zeile 25 in der Struktur gespeichert. In Zeile 26 werden dann noch geometrische Flächenparameter gesetzt.

```

27 (if(compute-object-fusion-p self)
28   (progn
29     (objects-fusion self nearest-object nr merging-type)
30     (update-hierarchie self root nearest-object))
31   (progn
32     (adjust-face abstract-copy face nr distance moving-type my-lambda)
33     (update-hierarchie self abstract-copy root)
34     (if (not respond
35         (3d-abstract-object! nearest-object
36           :root root
37           :exception-list exception-list
38           :respond t
39           :merge-which self
40           :merge-face nr
41           :merging-type merging-type
42           :moving-type moving-type
43           :my-lambda (- 1 my-lambda))))))...))

```

In Zeile 27 wird festgestellt, ob verschmolzen oder dimensioniert werden soll. In Zeile 29 werden die Objekte miteinander verschmolzen, und anschließend (Zeile 30) wird die Abstraktionshierarchie angepaßt. In den Zeilen 31-33 wird das Objekt dimensioniert, indem die entsprechende Seite verschoben wird. In Zeile 34 wird zunächst getestet, ob es sich schon um einen rekursiven Aufruf handelt. Ist dies nicht der Fall, so wird ein rekursiver Aufruf durchgeführt (Zeile 35-43). Dabei ist zu beachten, daß dies auch den Aufruf einer anderen Version der Methode (z.B. für den Zylinder) bewirken kann. *Respond* wird nun auf *true* gesetzt. Die durch die Regeln ermittelten Abstraktionsparameter werden nun übergeben (Zeile 39-43). Der Lambdafaktor wird gewonnen, indem die Differenz zu eins gebildet wird.

²Z.B.: Gibt es überhaupt irgendwelche benachbarten Flächen oder gehören diese zu Objekten, die in der `exception-list` stehen?

7.3.2 Projektionsabstraktion

Die Implementation der Abstraktion auf der Projektionsebene gestaltet sich wesentlich einfacher, als die auf der Modellebene. Die Abstraktion entsteht nicht erst in einem iterativen Prozeß, sondern wird schon während der Ermittlung des lokalen Kontextes bestimmt.

Anhand der Methode für den Quader wird die 2D-Abstraktion erläutert.

```
1(defmethod (2d-abstract-object abstract-quader)
2      (stream &key
3          (dashvector '#(3 3))
4          (thickness 0))
5      )
6  (let ((body-display (body-display-item sgeo-object)))
7      (loop for edge in (collect-drawn-edges body-display)
8          for tail = (xy-coords body-display (tail-of-edge edge))
9          for head = (xy-coords body-display (head-of-edge edge))
10         do (Progn
11             (draw-line (first tail) (second tail)
12                       (first head) (second head)
13                       :stream stream
14                       :thickness thickness)
15             (draw-line (first tail) (second tail)
16                       (first head) (second head)
17                       :alu :erase
18                       :dashed (check-whether-dashed-p)
19                       :dash-pattern dashvector
20                       :stream stream
21                       :draw-partial-dashes nil
22                       :thickness thickness))))))
```

Die Variable *Stream* entspricht dem Ausgabemedium (Zeile 1). In Zeile 3 und 4 werden die Abstraktionsparameter gesetzt. Die Variable *body-display* wird in Zeile 6 an die Kantenrepräsentation des geometrischen Objekts gebunden. Ab Zeile 7 wird die Abstraktion an der Kantenrepräsentation vorgenommen. Dabei wird zunächst die Darstellung der Kante gelöscht und dann erst wird sie abstrahiert gezeichnet.

7.4 Testumgebung

Ob eine Abstraktion ihren Zweck erfüllt und als solche überhaupt von einem Betrachter akzeptiert und erkannt wird, ist sehr schwer festzustellen. Denn Abstraktionen existieren nicht in der realen Welt; nur die Imaginationskraft des Menschen und der entsprechende Kontext ermöglichen eine korrekte Interpretation. Eine Testumgebung ist deshalb unbedingt notwendig, um Abstraktionsergebnisse zu überprüfen und Abstraktionsregeln zu verfeinern. Für das System PROXIMA steht eine solche Testumgebung zur Verfügung.

7.4.1 Ladeanweisung

Im Lisp-Listener von Genera 8.0.1 wird mit dem Befehl **Load System gd-abstract** das Gesamtsystem geladen³. Anschließend kann man mit dem Aufruf (**Start-Proxima**) die Oberfläche starten.

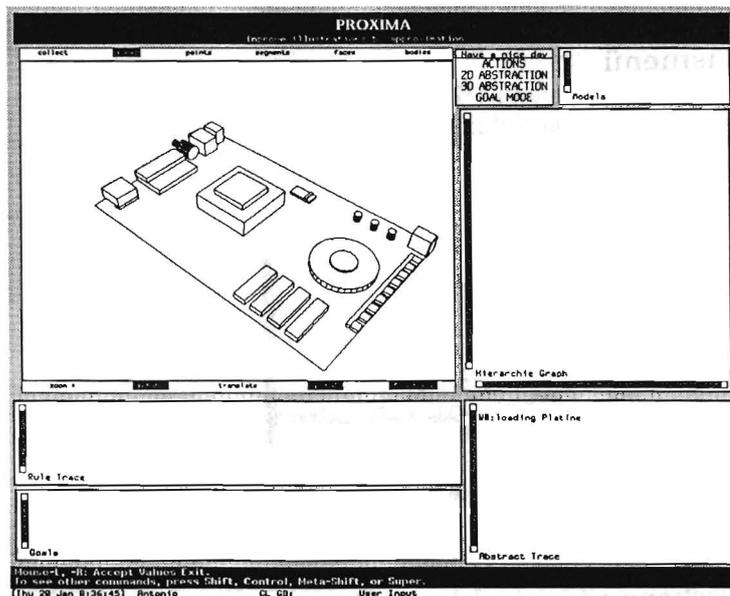


Abbildung 7.1: Die Testumgebung für das System PROXIMA.

³Wurde die Lispmaschine mit dem Boot-File **Diplom.boot.newest** hochgefahren, so ist das System PROXIMA bereits in der Welt vorhanden.

7.4.2 Fensterlayout

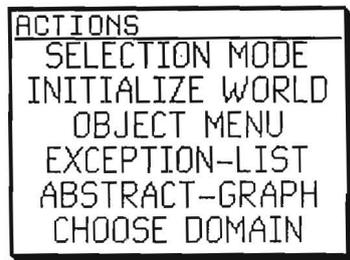
Die Geometrie der Oberfläche ist in Abbildung 7.1 zu sehen. Das große Fenster oben links ist das *Graphikausgabefenster*, darunter befindet sich das Ausgabefenster für den Regelinterpreter und ganz unten ein Fenster in dem die Abstraktionsspezifikationen abgebildet werden. Das kleine Fenster oben rechts dient zur Kontrolle der von PROXIMA neu generierten S-Geometry Objekte. Das Fenster darunter dient als Ausgabemedium der Abstraktionshierarchie, die als Graph visualisiert werden kann. Im Fenster ganz unten rechts wird der gesamte Abstraktionsprozeß dokumentiert.

7.4.3 Die Menüs

In der Mitte befindet sich das Hauptmenü, über das alle Funktionen angesteuert werden können. Es enthält die Menüpunkte *Actions*, *2D-Abstraction*, *3D-Abstraction* und *Goal Mode*.

7.4.3.1 Aktionsmenü

Über den Menüpunkt *Actions* gelangt man in das Aktionsmenü.



Das Aktionsmenü

Hier kann der *Selectionmode* geändert werden, der darüber entscheidet, ob die Auswahl der Objekte interaktiv⁴ oder menügesteuert erfolgen soll. Mit *Initialize World* können Abstraktionen wieder rückgängig gemacht werden. Mithilfe von *Object Menu* können Objekte menügesteuert ausgewählt werden.

⁴Dabei können die zu abstrahierenden Objekte direkt im Graphikfenster mit der Maus selektiert werden.

Objekte, die nicht abstrahiert werden sollen, werden über *Exception List* markiert. Die Visualisierung der Abstraktionshierarchie wird über den Punkt *Abstract Graph* aktiviert bzw. deaktiviert. Die Modelldomäne⁵ wird über den letzten Menüpunkt (*Choose Domain*) gewählt.

7.4.3.2 2D-Abstraktion

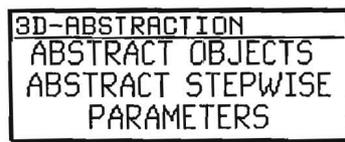
Wählt man diesen Menüpunkt, so gelangt man in ein Submenü zur Projektionsabstraktion.



2D-Abstraktionsmenü

Hier können zusätzlich die *Parameter* Strichstärke und Strichabstände verändert werden.

7.4.3.3 3D-Abstraktion



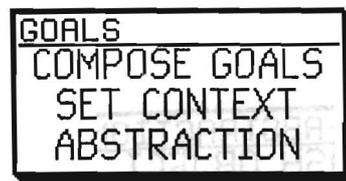
3D-Abstraktionsmenü

Parameters erlaubt die Modifikation der Abstraktionsparameter. Der Abstraktionsvorgang kann durch *Abstract stepwise* schrittweise oder durch *Abstract objects* bis zum sichtbaren Limit angehoben werden.

⁵Es stehen zur Zeit die Modemplatine und eine einfachere Testwelt zur Verfügung.

7.4.3.4 Zielmodus

Über *Goals Mode* gelangt man in den vollautomatischen Betriebsmodus von PROXIMA. Die Darstellungsbedingungen können durch *Compose Goals* festgelegt werden. Durch *Set Context* wird anschließend die Propagierung der Bedingungen durchgeführt. Der Menüpunkt *Abstraction* startet den vollautomatischen Abstraktionsvorgang.



Das Zielmenü

7.5 Programmierschnittstelle

Dieser Abschnitt enthält eine Auflistung der wichtigsten Schnittstellenfunktionen, Methoden und globalen Variablen, durch die Proxima unabhängig von der Oberfläche angesteuert werden kann.

7.5.1 Globale Variablen

- | | |
|----------------------------------|---|
| *abstract-collected-list* | enthält die Objekte, die abstrahiert werden sollen. |
| *abstract-exception-list* | enthält Objekte, die nicht abstrahiert werden sollen. |
| *abstract-dash-vector* | enthält einen Vektor der Form (X Y), wobei X die Strichlänge und Y die Lückengröße sind. Wird bei der Projektionsabstraktion verwendet. |
| *abstract-back-thickness* | bestimmt die Strichstärke der Hintergrundobjekte. |

abstract-fore-thickness	bestimmt die Strichstärke der Vordergrundobjekte.
db	enthält die Datenbasis des Regelinterpreters.
ri	enthält die aktuelle Instanz des Regelinterpreters.

7.5.2 Initialisierung der Abstraktionshierarchie

Mit dem Funktionsaufruf

(init-hierarchie-structure-from-scene SGEO-OBJEKTE)

wird die Abstraktionshierarchie initialisiert. SGEO-OBJEKTE bezeichnet eine Liste mit S-Geometry Objekten, die in den Abstraktionsprozeß miteinbezogen werden sollen. Die Verbindung zwischen S-Geometry Objekten und Abstraktionsobjekten wird mit der Funktion

(find-abstract-object SGEO-OBJECT)

ermöglicht.

7.5.3 Interaktiver Betriebsmodus

In diesem Betriebsmodus müssen zunächst alle relevanten globalen Variablen gesetzt werden. Dies sind:

abstract-back-thickness
abstract-fore-thickness
abstract-collected-list
abstract-exception-list
abstract-dash-vector

Nun kann durch die Methode

`(wb-merge-abstract-objects)` auf der Modellebene

und mit der Funktion

`(dash-choose-objects)` auf der Projektionsebene abstrahiert werden.

7.5.4 Vollautomatischer Betriebsmodus

Der Regelinterpreter und die Datenbasis werden durch den Befehl

`(init-rules)`

initialisiert.

Die Darstellungsbedingungen können durch den Befehl

`(cl::add-new-assertion '(visibility-constraints OBJECT CONSTRAINT) *db*)`

in die Datenbasis eingefügt werden. Dabei steht OBJECT für das zu abstrahierende Objekt und CONSTRAINT für eine der Darstellungsbedingungen **identifiable**, **classifiable**, **discriminable**, **realisable** oder **visible**.

Vorder- und Hintergrundobjekte können durch

`(cl::add-new-assertion '(Foreground-object OBJECT) *db*)`
`(cl::add-new-assertion '(Background-object OBJECT) *db*)`

gesetzt werden und das grundlegende Abstraktionsniveau wird durch

`(cl::add-new-assertion '(default-abstraction-level LEVEL) *db*)`

festgelegt. LEVEL steht für eine der drei Ebenen *High*, *Medium* und *Low*.

Der Kontext wird durch

`(set-context6 *ri* *db* '((merge-request OBJECT)))`
`(link-constraints-to-abstract-objects)`

gesetzt. Danach enthält die Datenbasis alle Darstellungsbedingungen, die für die automatische Abstraktion benötigt werden.

Abstrahiert wird dann mit den Methoden

`(abstract-up! OBJECT)` für die Modellabstraktion, bzw.
`(engage-2d-abstraction)` für die Projektionsabstraktion.

Soll der Abstraktionsgrad schrittweise angehoben werden, so geschieht dies durch die Methode

`(engage-3d-abstraction-stepwise)`

⁶Aus Konsistenzgründen wurde der Originalname *verfolge-ziel* umbenannt.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Erzielte Ergebnisse

In dieser Arbeit wurde ein Abstraktionssystem vorgestellt, das detailreiche Drahtrahmenmodelle in einfachere Modelle überführt. Entscheidende Verbesserungen und Vorteile gegenüber den bisherigen Arbeiten, die in Kapitel 3 vorgestellt wurden, sind:

- Der dynamische Abstraktionsprozeß:
Abstraktionen werden erst auf Anfrage generiert und können so auf die speziellen Bedürfnisse zugeschnitten werden. Damit verbunden ist auch die Formulierung der Abstraktionsaufgabe durch Darstellungsbedingungen. So ist es unkompliziert möglich, Anforderungen an das Abstraktionsergebnis zu stellen.
- Der iterative Abstraktionsprozeß:
Die schrittweise Anhebung des Abstraktionsgrads ermöglicht eine gute Kontrolle des Abstraktionsresultats. Darstellungsbedingungen bleiben in jedem Schritt erfüllt.
- Der komplexe Abstraktionsprozeß:
Durch die Kombination von 3D- und 2D-Techniken lassen sich vielfältige Abstraktionsergebnisse realisieren.

Die wenigen Vorarbeiten auf diesem Gebiet machten eine gründliche Recherche von Grundlagen und Bildbeispielen notwendig. Eine Implementierung aller Details, die im theoretischen Teil dieser Arbeit (speziell in Kapitel 2) vorgestellt wurden, hätte den Rahmen einer Diplomarbeit bei weitem überschritten. Aus diesem Grund bleibt noch viel Raum für Erweiterungen und Verbesserungen, von denen im folgenden nur die wichtigsten genannt werden.

8.2 Erweiterungsvorschläge

Das in Kapitel 4 und 5 vorgestellte Abstraktionskonzept, auf dem PROXIMA basiert, kann noch ergänzt werden.

8.2.1 Verfeinerung von Abstraktionszielen und Randbedingungen

In Kapitel 2 wurden neben den Abstraktionstypen, die für die Realisierung von Abstraktionen zuständig sind, auch die Abstraktionsziele angeführt, die mit Abstraktionen verfolgt werden können.

Es wurde nur anhand von Beispielen erläutert, wie diese Ziele in Darstellungsbedingungen an einzelne Objektteile formuliert werden können. Eine genauere Betrachtung, wie Randbedingungen (siehe Abschnitt 2.4) die Verfolgung von Abstraktionszielen beeinflussen, wurde nicht durchgeführt. Durch eine weitere Untersuchung dieses Themenkomplexes kann die Flexibilität des Abstraktionssystems, auf einen wechselnden Kontext adäquat und selbständig zu reagieren, erhöht werden. So können dann auch von exemplarisch gespeicherten Modellen prototypische Vertreter generiert werden, wie sie in Abschnitt 2.2 beschrieben werden.

8.2.2 Weitere Abstraktionstypen

In PROXIMA finden die Abstraktionstypen *Verschmelzen*, *Verändern der Dimensionen*, *Weglassen* und *Vereinheitlichen* Verwendung. Darüber hinaus stellen weitere Abstraktionstypen, die in Abschnitt 2.3 vorgestellt wurden, eine sinnvolle Ergänzung dar. Vor allem das *Symbolisieren* wird häufig in technischen Dokumentationen verwendet. Denkbar wäre, daß das System auf eine

Bibliothek solcher Symbole zurückgreift, um durch einen Ersetzungsvorgang abstrahierte Darstellungen zu generieren.

Aber auch der Abstraktionstyp *Skalieren* würde variabelere Abstraktionsresultate ermöglichen. Durch eine „Lupenfunktion“ könnten Objekte in den Vorder- bzw. Hintergrund gebracht werden.

Die Berücksichtigung von Renderattributen, Lichtquellen und Farben findet zur Zeit nicht statt¹. Farben und Struktureigenschaften könnten durch Mischen oder durch die dominanteste Eigenschaft im Abstraktionsergebnis Berücksichtigung finden. Die Verwendung dieser Attribute in der Abstraktion führt zu einer Reihe von Effekten, die eine variabelere Gestaltung des Abstraktionsresultats ermöglichen (z.B. Fokussierung durch einen „Spot“). Die Untersuchung, wie welche Attribute in abstrahierten Objekten gemischt werden, läßt im Hinblick auf mächtigere Hardware-Plattformen eine deutliche Verbesserungen der Bildqualität erwarten.

Im Vergleich zur 3D-Abstraktion wurden die 2D-Abstraktionsalgorithmen nicht so ausführlich behandelt. Viele Techniken der Modellabstraktion (z.B. das *Verschmelzen*) lassen sich aber auch auf die Projektionsebene übertragen.

8.2.3 Automatische Evaluierung generierter Abstraktionen

Eine automatische Auswertung der Abstraktionen würde dazu beitragen die Abstraktionsresultate weiter zu verbessern. Zur Realisierung bietet sich z.B. ein funktional basiertes Objekterkennungssystem an. Wie in Abschnitt 3.2.4 erläutert wurde, prüft ein solches System anhand funktionaler Eigenschaften die Kategoriezugehörigkeit eines Objekts. Als ungenügend klassifizierte Abstraktionen können dann vielleicht sogar mit einer Begründung zurückgewiesen werden, die vom Abstraktionssystem berücksichtigt werden kann.

8.2.4 Abstraktionsregeln

Zur Zeit greift PROXIMA auf ein recht einfaches Regelpaket zurück. Mit der Einführung neuer Abstraktionstypen müssen aber auch neue Abstraktionsregeln hinzugefügt werden, um eine sinnvolle Parametrisierung der Abstrak-

¹Es ist zwar möglich, gerenderte Bilder von Abstraktionen zu erzeugen, allerdings werden die Renderattribute zur Zeit von Hand gesetzt.

tionstypen zu realisieren.

Es ist auch denkbar, Randbedingungen durch ein Regelpaket in den Inferenzprozeß miteinzubeziehen. Dies könnte in Kombination mit Überlegungen, die aus Abschnitt 8.2.1 resultieren, eine interessante Weiterentwicklung bedeuten.

8.2.5 Signifikante Attribute

Die Repräsentation der signifikanten Attribute in der Form, wie sie in PROXIMA (siehe dazu Abschnitt 4.1.2), realisiert wurden, entsprang dem Wunsch, für eine kleine Domäne eine einfache und effiziente Lösung zu erhalten.

Eine weitere Entwicklung könnte darin resultieren, diese Repräsentation mehr einer prototypischen Repräsentation anzupassen. Denkbar wäre z.B. das Speichern solcher Information in KRIS/RAT (vergl. auch [28]). Damit könnte PROXIMA auf schon bestehende Wissensrepräsentationen zurückgreifen.

Literaturverzeichnis

- [1] Ach, N., Über die Begriffsbildung, Bamberg: Buchner, 1921.
- [2] Allen, L., YAPS: Yet Another Production System. Tech. Report No. TR-11-46, Computer Science Dept., University of Maryland, 1982.
- [3] Anderson, J.R., Kognitive Psychologie: Eine Einführung, Heidelberg: Spektrum der Wissenschaft, 1988.
- [4] André, E., Rist, T., The Design of Illustrated Documents as a Planning Task, in: M. Maybury (Hrsg.), Intelligent Multimedia Interface, AAAI/MIT Press, 1993.
- [5] Autorenkollektiv, AutoCAD 12, Product Description, Autodesk AG, 1991.
- [6] Backe, H., König, L., Elektrotechnik und Elektronik selbst erlebt, Berlin: Urania-Verlag, 1979.
- [7] Berlyne, D.E., Aesthetics and Psychobiology, New York: Appleton-Century-Crofts-Educational Division, 1971.
- [8] Bies, C., Mitschang, B., Mitschang, P., Beanspruchungsgerechtes Konstruieren, Informatik Forschung und Entwicklung 9, S. 201-212, 1994.
- [9] Bolz, D., Some Aspects of the User Interface of a Knowledge Based Beautifier for Drawings, in: Proc. of the International Workshop on Intelligent User Interfaces in Orlando, S. 45-51, 1993.
- [10] Der große Brockhaus in 12 Bänden, Wiesbaden: Brockhaus, 1977.

- [11] Brooks, L.R., Nonanalytic concept formation and memory for instances, in: E. Rosch, B.B. Loyd, *Cognition and Categorization*, S. 169-221, NJ: Erlbaum, 1978.
- [12] Butz, A., BETTY: Planung und Generierung von Animationssequenzen zur Visualisierung von Bewegungen und räumlichen Zusammenhängen, Diplomarbeit, Universität des Saarlandes, Fachbereich für Informatik. Prof. Dr. W. Wahlster, 1994.
- [13] Cawkell, A.E., Indexing and retrieving graphical images, *CRITIQUE* 3(5), S. 1-12, 1991.
- [14] Cooper T., Worgin, N., *Rule-based Programming with OPS5*, Morgan Kaufmann, 1988.
- [15] Dumont's Chronik der Kunst im 20. Jahrhundert : Stile, Akteure und Meisterwerke der Moderne von Jean-Louis Ferrier (Hrsg.), Köln: DuMont, 1990.
- [16] Autorenkollektiv, *Produktinformation: Linde Polaris 2603 ADL, Duo-Frost Kühl- und Gefriergeräte GmbH*, 1978.
- [17] Eckes, T., *Psychologie der Begriffe*, Göttingen: Hogrefe Verlag, 1991.
- [18] Emhardt, J., Strothotte, T., Hyper-Rendering, in: *Proc. Graphics Interface '92 in Vancouver*, S. 43-47, 1992.
- [19] Endres-Niggemeier, B., Hobbs, J., Sparck Jones, K., (Eds.), *Summarizing Text for Intelligent Communication*, Dagstuhl Seminar Report 79 (9350), 1993.
- [20] Feiner, S., APEX: An Experiment in the Automated Creation of Pictorial Explanations, in: *IEEE Computer Graphics and Applications* 5(11), S. 117-123, 1985.
- [21] Feiner, S., Seligman, D., Automated Generation of Intent-Based 3D-Illustrations, *Computer Graphics*, 25(3), 1991.
- [22] Fellner, W., *Computer Grafik*, Wissenschaftsverlag, 1988.

- [23] Fisher, S.C., The process of generalizing abstraction; and its product the general concept, *Psychological Monographs*, 21 (Whole No.90), 1916.
- [24] Foley, J., van Dam, A., Hughes, J., Feiner, S., *Computer Graphics: Principles and Practice* (2nd Edition), Addison-Wesley, 1990.
- [25] Galton, F., *Inquiries into human faculty and its development*, London: Macmillan, 1883.
- [26] Görz, G., (Hrsg.), *Einführung in die künstliche Intelligenz*, Addison-Wesley (Deutschland), 1993.
- [27] Graf, W., LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations, in: Salvendy, G. Smith, M.J. (Hsg.), *Proc. of the 5th International Conference on Human-Computer Interaction (jointly with 9th Symposium on Human Interfaces, Japan)*, 1993.
- [28] Heinsohn, J., Kudenko, D., Nebel, B., Profitlich, H.J., RAT: Representation of Actions using Terminological Logics, in: *Proc. of the DFKI Workshop on Taxonomic Reasoning*, DFKI Document D-92-08, DFKI GmbH, Saarbrücken, 1992.
- [29] Henne, P., Schmitgen, G., *Graphisches Suchsystem SKAGRA – Konzepte und Realisierung*, Tasso-Report Nr. 49, GMD, 1993.
- [30] Hollerbach, J., *Hierarchical Shape Description of Objects by Selection and Modifications of Prototypes*, MIT Dept. of Elec. Eng., MS Thesis, AI-TR-346, 1975.
- [31] Hull, C.L., *Quantitative aspects of the evolution of concepts*, *Psychological Monographs*, 28 (Whole No. 123), 1920.
- [32] Jolicoeur, P., Gluck, A., *Pictures and Names: Making the Connection*, in: *Cognitive Psychology* 16, S. 243-275, 1984.
- [33] Katz, D., *Gestaltpsychologie*, bearbeitet und ergänzt von W. Metzger, M. Stadler und H. Crabus, 4. Auflage, Basel: Schwabe, 1969.
- [34] Kilger, A., Finkler, W., TAG-Based Inkremental Generation, in: *Computational Linguistics*, MIT Press, to appear 1995.

- [35] Koehler, J., Correct modification of complex plans, in: Proc. of the 11th Conference on Artificial Intelligence, S. 605-609, Amsterdam: John Wiley, 1994.
- [36] Kolla, R., Molitor, P., Osthof, H. G., Einführung in den VLSI-Entwurf, Stuttgart: B.G. Teubner, 1989.
- [37] Krüger, A., Schneider, G., Schneiderlöchner, F., Schommer, C., Wissensbasierte Graphikgenerierung, Abschlußbericht Fortgeschrittenen Praktikum, Universität des Saarlandes, Fachbereich für Informatik, Lehrstuhl Prof. Dr. W. Wahlster, 1990.
- [38] Kurzidim, M., Bildersafari, in: c't 9/94, S. 112-121, Heise Verlag, 1994.
- [39] Labov, W., The boundaries of words and their meanings, in: C.J.N. Bailey & R. W. Shuy (Eds.), New ways of analyzing variation in English, S. 340-373, Washington D.C.: Georgetown University Press, 1973.
- [40] Locke, J., An essay concerning human understanding, in: British empirical philosophers: Locke, Berkley, Hume, Reid and J.S. Mill, S. 31-160, London: Routledge & Kegan, 1952.
- [41] Malet, R.M., Joan Miró, Stuttgart: Klett-Cotta, 1984.
- [42] Mathaei, J.M., Grundfragen des Graphik-Design, Heinz Moos Verlag, 1975.
- [43] Mervis, C.B., Rosch, E., Categorization of Natural Languages, in: Ann. Rev. Psychology (32), S. 89-115, 1989.
- [44] Minsky, M., A framework for representing knowledge, in: P. H. Winston (Hrsg.), The psychology of computer vision, S. 211-27, New York: McGraw-Hill, 1975.
- [45] Müller, J., GeoDisplay: Ein flexibles 3D-Visualisierungswerkzeug für Lisp, Diplomarbeit, Universität des Saarlandes, Fachbereich für Informatik, Lehrstuhl Prof. Dr. W. Wahlster, 1995.
- [46] Autorenkollektiv, Tipmatic 1019, Pfaff Handelsgesellschaft für Haushaltsgeräte mbH, 1982.

- [47] Autorenkollektiv, Bedienungsanleitung: Café Gourmet, Philips AG, 1989.
- [48] Plastock, R.A., Kalley, G., Computergrafik, New York: McGraw-Hill, 1985.
- [49] Rist, T., André, E., Wissensbasierte Perspektivenwahl für die automatische Erzeugung von 3D-Objektdarstellungen, in: Klaus Kansky und Peter Wißkirchen (Hrsg.), Graphik und KI, IFB 239, Berlin: Springer-Verlag, 1990.
- [50] Rist, T., André, E., Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP, in: Proc. AVI'92, World Scientific, 1992.
- [51] Rist, T., Krüger, A., Schneider, G., Zimmermann, D., AWI: A Workbench for Semi-Automated Illustration Design, in: Proc. AVI '94, ACM Press, 1994.
- [52] Rist, T., Ein wissensbasierter Ansatz zum automatischen Entwurf von Gebrauchsgraphiken, in Vorbereitung.
- [53] Schneider, G., Eine Werkbank zur Erzeugung von 3D-Illustrationen, Diplomarbeit, Universität des Saarlandes, Fachbereich für Informatik, Lehrstuhl Prof. Dr. W. Wahlster, 1995.
- [54] Schneider, G., Krüger, A., TOPAS: Eine Werkbank zur Erzeugung von 3D-Illustrationen, in: A. Iwainsky (Hrsg.) , Workshop auf der Wartburg, Computergrafik und automatisierte Layoutsynthese, S. 115-133, 1993.
- [55] Schneiderlöchner, F., Gordian: Generierung von Referenzausdrücken in multimodalen Dokumenten, Diplomarbeit, Universität des Saarlandes, Fachbereich für Informatik, Lehrstuhl Prof. Dr. W. Wahlster, in Vorbereitung.
- [56] Autorenkollektiv, Operating Instructions: Compact Disc Compact Player, Sony Corporation, 1990.
- [57] Stark, L., Bowyer, K., Function-Based Object Recognition for Multiple Object Categories, in: Advances in Structural and Syntactic Pattern Recognition, H. Bunke (Hrsg), World Scientific, 1992.

- [58] Steele, G. L., Common Lisp - The Language, Digital Press, 1990.
- [59] Autorenkollektiv, Betriebs- und Serviceanleitung, Sursee Gretner AG, 1989.
- [60] Common Lisp Programming Constructs, Symbolics Inc., Burlington (USA), 1990.
- [61] Genera User's Guide, Symbolics Inc., Burlington (USA), 1990.
- [62] S-Geometry, Symbolics Inc., Burlington (USA), 1990.
- [63] The Knowledge Engineering Tool S.1., Teknowledge Inc., Palo Alto (USA), 1984.
- [64] Wahlster, W., André, E., Finkler, W., Profitlich, H.-J., Rist, T., Plan-based integration of natural language and graphics generation, in: Artificial Intelligence 63, Elsevier Science Publisher, S. 387-427 1993.
- [65] Wahlster, W., André, E., Bandyopadhyay, E., Graf, W., Rist, T., WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation, in: Ortony, A., Slack, J., Stock, O. (Hrsg.), Communication from an Artificial Intelligence Perspective: Theoretical and Applied Issues, S. 121-144, Berlin: Springer-Verlag, 1992.
- [66] Wittgenstein, L., Philosophische Untersuchungen, Frankfurt: Suhrkamp, 1971.
- [67] Wootton, R.J., Das Design von Insektenflügeln, in: Spektrum der Wissenschaft 1/91, S.61, 1991.
- [68] Zimmermann, D., AnnA II: Ein wissensbasiertes System zur automatischen Annotation von Grafiken, Diplomarbeit, Universität des Saarlandes, Fachbereich für Informatik, Prof. Dr. W. Wahlster, 1993 .

Abbildungsverzeichnis

1.1	Joan Miró: <i>Gemälde nach einer Collage</i> (1933)	3
1.2	Vorbereitende Collage für das Gemälde in Abb. 1.1	3
1.3	Die vergrößerte Darstellung eines Libellenflügelstücks.	4
1.4	Die abstrahierte Darstellung des Libellenflügelstücks.	5
2.1	Der Schraubenzieher als prototypischer Vertreter.	10
2.2	Unwichtige Teile der Nähmaschine wurden <i>weggelassen</i>	11
2.3	Der Abstraktionstyp <i>Verschmelzen</i>	12
2.4	Der Abstraktionstyp <i>Dimensionieren</i>	13
2.5	Strichstärken werden vereinheitlicht.	13
2.6	Bildinhalte werden skaliert.	14
2.7	Die symbolische Darstellung eines Schaltkreises.	14
2.8	Konsistente Abstraktion innerhalb eines Bildes.	15
2.9	Persistenzbedingungen müssen beachtet werden.	16
3.1	Beispiel zur Vagheit linguistischer Kategorien.	21
3.2	Die graphische Abstraktion einer Knopfleiste.	28
3.3	Ein Hierarchiebaum mit dem dazugehörigen Abstraktionsbaum.	31
3.4	Die Lücken zwischen den Quadern werden durch „Wachsen“ geschlossen.	32
3.5	Eine Quaderreihe wird zu zwei neuen Objekten verschmolzen.	33
3.6	Relativ kleine Objekte werden eliminiert.	34
3.7	Flache und dünne angrenzende Objekte werden entfernt.	34
3.8	Durch Abstraktion der Knopfleiste wird Knopf 1 hervorgehoben.	35
3.9	Der dunkle Würfel wird durch die dünne Fläche verdeckt.	36
3.10	Das Entfernen der dünnen Fläche kann zu verwirrenden Ergebnissen führen.	36

3.11	Eine mit dem Skizzen-Renderer realisierte Graphik.	37
3.12	Die Begriffshierarchie und die Funktionalität der Objekte. . .	40
3.13	Verschönerung einer Graphik	42
4.1	Grundlegendes Schema für ein Abstraktionssystem.	45
4.2	Signifikante Attribute eines Sportwagen.	49
4.3	Signifikante Attribute müssen berücksichtigt werden.	50
4.4	Das vervollständigte Abstraktionsschema.	53
5.1	Die drei Grundprimitive: Zylinder, Quader und Lamina. . . .	55
5.2	Die Auswirkungen der Darstellungsbedingungen der IC-Bank auf die Darstellungsbedingungen der vier ICs.	58
5.3	Die LED-Reihe wird abstrahiert.	60
5.4	Wie wirkt sich die Spezifikation auf den lokalen Kontext aus?	64
5.5	Die Parametrisierung des Abstraktionstyps <i>Wachsen</i>	68
5.6	Der Abstraktionstyp <i>Verschmelzen</i> ist parametrisierbar.	69
5.7	Zwei Beispiele zum <i>Verschmelzen</i>	71
5.8	Ein Würfel und ein Zylinder werden verschmolzen	72
5.9	Beispiele für die Projektionsabstraktion	75
6.1	Das Modell einer Modemplatine als Testdomäne.	77
6.2	Interaktive Abstraktion.	78
6.3	Die Architektur des WIP-Systems.	80
6.4	Abstraktionssequenz Modellebene und Projektionsebene	83
6.5	Eine Kombination der Illustrationstechniken Abstraktion, Pfeil und Inset.	84
6.6	Eine gerenderte Abstraktionssequenz	85
6.7	Abstrahieren mit AWI	87
6.8	GORDIAN referenziert eine Abstraktion.	90
6.9	Über einem Bestand von Graphiken kann mithilfe der Ab- straktion eine Suchhierarchie erzeugt werden.	91
7.1	Die Testumgebung für das System PROXIMA.	102



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

-Bibliothek, Information und Dokumentation (BID)- PF 2080 6760S Kaiserslautern FRG	Telefon (0631) 205-350 Telefax (0631) 205-321 e-mail dfkibib@dfki.uni-kl.de WWW http://www.dfki.uni- sb.de/dfkibib
--	--

Veröffentlichungen des DFKI

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder (so sie als per ftp erhaeltlich angemerkt sind) per anonymous ftp von ftp.dfki.uni-kl.de (131.246.241.100) im Verzeichnis pub/Publications bezogen werden. Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or (if they are marked as obtainable by ftp) by anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) in the directory pub/Publications.

The reports are distributed free of charge except where otherwise noted.

DFKI Research Reports

1995

RR-95-11

Anne Kilger, Wolfgang Finkler
Incremental Generation for Real-Time Applications
47 pages

RR-95-09

M. Buchheit, F. M. Donini, W. Nutt, A. Schaefer
A Refined Architecture for Terminological Systems:
Terminology = Schema + Views
71 pages

RR-95-07

Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt
The Complexity of Concept Languages
57 pages

RR-95-04

M. Buchheit, H.-J. Bürckert, B. Hollunder, A. Laux, W. Nutt, M. Wójcik
Task Acquisition with a Description Logic Reasoner
17 pages

RR-95-03

Stephan Baumann, Michael Malburg, Hans-Guenther Hein, Rainer Hoch, Thomas Kieninger, Norbert Kuhn
Document Analysis at DFKI
Part 2: Information Extraction
40 pages

RR-95-02

Majdi Ben Hadj Ali, Frank Fein, Frank Hoenes, Thorsten Jaeger, Achim Weigel
Document Analysis at DFKI
Part 1: Image Analysis and Text Recognition
69 pages

1994

RR-94-39

Hans-Ulrich Krieger
Typed Feature Formalisms as a Common Basis for Linguistic Specification.
21 pages

RR-94-38

Hans Uszkoreit, Rolf Backofen, Stephan Busemann, Abdel Kader Diagne, Elizabeth A. Hinkelman, Walter Kasper, Bernd Kiefer, Hans-Ulrich Krieger, Klaus Netter, Günter Neumann, Stephan Oepen, Stephen P. Spackman.
DISCO-An HPSG-based NLP System and its Application for Appointment Scheduling.
13 pages

RR-94-37

Hans-Ulrich Krieger, Ulrich Schäfer
TDL - A Type Description Language for HPSG, Part 1: Overview.
54 pages

- RR-94-36**
Manfred Meyer
 Issues in Concurrent Knowledge Engineering. Knowledge Base and Knowledge Share Evolution.
 17 pages
- RR-94-35**
Rolf Backofen
 A Complete Axiomatization of a Theory with Feature and Arity Constraints
 49 pages
- RR-94-34**
Stephan Busemann, Stephan Oepen, Elizabeth A. Hinkelmann, Günter Neumann, Hans Uszkoreit
 COSMA – Multi-Participant NL Interaction for Appointment Scheduling
 80 pages
- RR-94-33**
Franz Baader, Armin Laux
 Terminological Logics with Modal Operators
 29 pages
- RR-94-31**
Otto Kühn, Volker Becker, Georg Lohse, Philipp Neumann
 Integrated Knowledge Utilization and Evolution for the Conservation of Corporate Know-How
 17 pages
- RR-94-23**
Gert Smolka
 The Definition of Kernel Oz
 53 pages
- RR-94-20**
Christian Schulte, Gert Smolka, Jörg Würtz
 Encapsulated Search and Constraint Programming in Oz
 21 pages
- RR-94-18**
Rolf Backofen, Ralf Treinen
 How to Win a Game with Features
 18 pages
- RR-94-17**
Georg Struth
 Philosophical Logics—A Survey and a Bibliography
 58 pages
- RR-94-16**
Gert Smolka
 A Foundation for Higher-order Concurrent Constraint Programming
 26 pages
- RR-94-15**
Winfried H. Graf, Stefan Neurohr
 Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces
 20 pages
- RR-94-14**
Harold Boley, Ulrich Buhrmann, Christof Kremer
 Towards a Sharable Knowledge Base on Recyclable Plastics
 14 pages
- RR-94-13**
Jana Koehler
 Planning from Second Principles—A Logic-based Approach
 49 pages
- RR-94-12**
Hubert Comon, Ralf Treinen
 Ordering Constraints on Trees
 34 pages
- RR-94-11**
Knut Hinkelmann
 A Consequence Finding Approach for Feature Recognition in CAPP
 18 pages
- RR-94-10**
Knut Hinkelmann, Helge Hintze
 Computing Cost Estimates for Proof Strategies
 22 pages
- RR-94-08**
Otto Kühn, Björn Höfling
 Conserving Corporate Knowledge for Crankshaft Design
 17 pages
- RR-94-07**
Harold Boley
 Finite Domains and Exclusions as First-Class Citizens
 25 pages
- RR-94-06**
Dietmar Dengler
 An Adaptive Deductive Planning System
 17 pages
- RR-94-05**
Franz Schmalhofer, J. Stuart Aitken, Lyle E. Bourne jr.
 Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse
 81 pages
- RR-94-03**
Gert Smolka
 A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards
 34 pages

RR-94-02

Elisabeth André, Thomas Rist

Von Textgeneratoren zu Intellimedia-Präsentationssystemen

22 Seiten

RR-94-01

Elisabeth André, Thomas Rist

Multimedia Presentations: The Support of Passive and Active Viewing

15 pages

1993

RR-93-48

Franz Baader, Martin Buchheit, Bernhard Hollunder

Cardinality Restrictions on Concepts

20 pages

RR-93-46

Philipp Hanschke

A Declarative Integration of Terminological, Constraint-based, Data-driven, and Goal-directed Reasoning

81 pages

RR-93-45

Rainer Hoch

On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition

21 pages

RR-93-44

Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, Martin Staudt

Subsumption between Queries to Object-Oriented Databases

36 pages

RR-93-43

M. Bauer, G. Paul

Logic-based Plan Recognition for Intelligent Help Systems

15 pages

RR-93-42

Hubert Comon, Ralf Treinen

The First-Order Theory of Lexicographic Path Orderings is Undecidable

9 pages

RR-93-41

Winfried H. Graf

LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations

9 pages

RR-93-40

Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf

Queries, Rules and Definitions as Epistemic Statements in Concept Languages

23 pages

RR-93-38

Stephan Baumann

Document Recognition of Printed Scores and Transformation into MIDI

24 pages

RR-93-36

Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck,

Ralf Legleitner, Gabriele Schmidt

Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld

13 Seiten

RR-93-35

Harold Boley, François Bry, Ulrich Geske (Eds.)

Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*

150 Seiten

Note: This document is available for a nominal charge of 25 DM (or 15 US-\$).

RR-93-34

Wolfgang Wahlster

VerbMobil Translation of Face-To-Face Dialogs

10 pages

RR-93-33

Bernhard Nebel, Jana Koehler

Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis

33 pages

RR-93-32

David R. Traum, Elizabeth A. Hinkelman

Conversation Acts in Task-Oriented Spoken Dialogue

28 pages

RR-93-31

Elizabeth A. Hinkelman, Stephen P. Spackman

Abductive Speech Act Recognition, Corporate Agents and the COSMA System

34 pages

RR-93-30

Stephen P. Spackman, Elizabeth A. Hinkelman

Corporate Agents

14 pages

RR-93-29

Armin Laux

Representing Belief in Multi-Agent Worlds via Terminological Logics

35 pages

RR-93-28

Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker
 Feature-Based Allomorphy
 8 pages

RR-93-27

Hans-Ulrich Krieger
 Derivation Without Lexical Rules
 33 pages

RR-93-26

Jörg P. Müller, Markus Pischel
 The Agent Architecture InteRRaP: Concept and Application
 99 pages

RR-93-25

Klaus Fischer, Norbert Kuhn
 A DAI Approach to Modeling the Transportation Domain
 93 pages

RR-93-24

Rainer Hoch, Andreas Dengel
 Document Highlighting — Message Classification in Printed Business Letters
 17 pages

RR-93-23

Andreas Dengel, Ottmar Lutz
 Comparative Study of Connectionist Simulators
 20 pages

RR-93-22

Manfred Meyer, Jörg Müller
 Weak Looking-Ahead and its Application in Computer-Aided Process Planning
 17 pages

RR-93-20

Franz Baader, Bernhard Hollunder
 Embedding Defaults into Terminological Knowledge Representation Formalisms
 34 pages

RR-93-18

Klaus Schild
 Terminological Cycles and the Propositional μ -Calculus
 32 pages

RR-93-17

Rolf Backofen
 Regular Path Expressions in Feature Logic
 37 pages

RR-93-16

Gert Smolka, Martin Henz, Jörg Würtz
 Object-Oriented Concurrent Constraint Programming in Oz
 17 pages

RR-93-15

Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster
 PLUS - Plan-based User Support Final Project Report
 33 pages

RR-93-14

Joachim Niehren, Andreas Podelski, Ralf Treinen
 Equational and Membership Constraints for Infinite Trees
 33 pages

RR-93-13

Franz Baader, Karl Schlechta
 A Semantics for Open Normal Defaults via a Modified Preferential Approach
 25 pages

RR-93-12

Pierre Sablayrolles
 A Two-Level Semantics for French Expressions of Motion
 51 pages

RR-93-11

Bernhard Nebel, Hans-Jürgen Bürckert
 Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra
 28 pages

RR-93-10

Martin Buchheit, Francesco M. Donini, Andrea Schaerf
 Decidable Reasoning in Terminological Knowledge Representation Systems
 35 pages

RR-93-09

Philipp Hanschke, Jörg Würtz
 Satisfiability of the Smallest Binary Program
 8 pages

RR-93-08

Harold Boley, Philipp Hanschke, Knut Hinkelmann, Manfred Meyer
 CoLAB: A Hybrid Knowledge Representation and Compilation Laboratory
 64 pages

RR-93-07

Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux
 Concept Logics with Function Symbols
 36 pages

RR-93-06

Hans-Jürgen Bürckert, Bernhard Hollunder, Armin Laux
 On Skolemization in Constrained Logics
 40 pages

RR-93-05

Franz Baader, Klaus Schulz
 Combination Techniques and Decision Problems for Disunification
 29 pages

RR-93-04

Christoph Klauck, Johannes Schwagereit
 GGD: Graph Grammar Developer for features in CAD/CAM
 13 pages

RR-93-03

Franz Baader, Bernhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, Enrico Franconi
 An Empirical Analysis of Optimization Techniques for Terminological Representation Systems
 28 pages

RR-93-02

Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, Thomas Rist
 Plan-based Integration of Natural Language and Graphics Generation
 50 pages

RR-93-01

Bernhard Hollunder
 An Alternative Proof Method for Possibilistic Logic and its Application to Terminological Logics
 25 pages

DFKI Technical Memos**1993****1995****TM-95-02**

Michael Sintek
 FLIP: Functional-plus-Logic Programming on an Integrated Platform
 106 pages

TM-93-05

Michael Sintek
 Indexing PROLOG Procedures into DAGs by Heuristic Classification
 64 pages

TM-95-01

Martin Buchheit, Rüdiger Klein, Werner Nutt
 Constructive Problem Solving: A Model Construction Approach towards Configuration
 34 pages

TM-93-04

Hans-Günther Hein
 Propagation Techniques in WAM-based Architectures — The FIDO-III Approach
 105 pages

1994**TM-94-04**

Cornelia Fischer
 PAnUDE – An Anti-Unification Algorithm for Expressing Refined Generalizations
 22 pages

TM-93-03

Harold Boley, Ulrich Buhrmann, Christof Kremer
 Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
 11 pages

TM-94-03

Victoria Hall
 Uncertainty-Valued Horn Clauses
 31 pages

TM-93-02

Pierre Sablayrolles, Achim Schupeta
 Conflict Resolving Negotiation for COoperative Schedule Management Agents (COSMA)
 21 pages

TM-94-02

Rainer Bleisinger, Berthold Kröll
 Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations
 11 pages

TM-93-01

Otto Kühn, Andreas Birk
 Reconstructive Integrated Explanation of Lathe Production Plans
 20 pages

TM-94-01

Rainer Bleisinger, Klaus-Peter Gores
 Text Skimming as a Part in Paper Document Understanding
 14 pages

DFKI Documents

1995

D-95-09

Antonio Krüger

PROXIMA: Ein System zur Generierung graphischer Abstraktionen
120 Seiten

D-95-07

Ottmar Lutzy

Morphic - Plus

Ein morphologisches Analyseprogramm für die deutsche Flexionsmorphologie und Komposita-Analyse
74 pages

D-95-06

Markus Steffens, Ansgar Bernardi

Integriertes Produktmodell für Behälter aus Faserverbundwerkstoffen
48 Seiten

D-95-05

Georg Schneider

Eine Werkbank zur Erzeugung von 3D-Illustrationen
157 Seiten

D-95-03

Christoph Endres, Lars Klein, Markus Meyer

Implementierung und Erweiterung der Sprache *ALCP*
110 Seiten

D-95-02

Andreas Butz

BETTY

Ein System zur Planung und Generierung informativer Animationssequenzen
95 Seiten

D-95-01

Susanne Biundo, Wolfgang Tank (Hrsg.)

Beiträge zum Workshop „Planen und Konfigurieren“, Februar 1995
169 Seiten

Note: This document is available for a nominal charge of 25 DM (or 15 US-\$).

1994

D-94-15

Stephan Oepen

German Nominal Syntax in HPSG
— On Syntactic Categories and Syntagmatic Relations
—
30 pages

D-94-14

Hans-Ulrich Krieger, Ulrich Schäfer

TDL - A Type Description Language for HPSG, Part 2: User Guide.
72 pages

D-94-12

Arthur Sehn, Serge Autexier (Hrsg.)

Proceedings des Studentenprogramms der 18. Deutschen Jahrestagung für Künstliche Intelligenz KI-94
69 Seiten

D-94-11

F. Baader, M. Buchheit, M. A. Jeusfeld, W. Nutt (Eds.)

Working Notes of the KI'94 Workshop: KRDB'94 - Reasoning about Structured Objects: Knowledge Representation Meets Databases
65 pages

Note: This document is no longer available in printed form.

D-94-10

F. Baader, M. Lenzerini, W. Nutt, P. F. Patel-Schneider (Eds.)

Working Notes of the 1994 International Workshop on Description Logics
118 pages

Note: This document is available for a nominal charge of 25 DM (or 15 US-\$).

D-94-09

Technical Staff

DFKI Wissenschaftlich-Technischer Jahresbericht 1993
145 Seiten

D-94-08

Harald Feibel

IGLOO 1.0 - Eine grafikunterstützte Beweisentwicklungsumgebung
58 Seiten

D-94-07

Claudia Wenzel, Rainer Hoch

Eine Übersicht über Information Retrieval (IR) und NLP-Verfahren zur Klassifikation von Texten
25 Seiten

D-94-06

Ulrich Buhrmann

Erstellung einer deklarativen Wissensbasis über recyclingrelevante Materialien
117 Seiten

D-94-04

Franz Schmalhofer, Ludger van Elst

Entwicklung von Expertensystemen: Prototypen, Tiefenmodellierung und kooperative Wissensentwicklung
22 Seiten

D-94-03*Franz Schmalhofer*

Maschinelles Lernen: Eine kognitionswissenschaftliche Betrachtung
54 Seiten

Note: This document is no longer available in printed form.

D-94-02*Markus Steffens*

Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff
90 pages

D-94-01*Josua Boon (Ed.)*

DFKI-Publications: The First Four Years
1990 - 1993
75 pages

1993**D-93-27***Rolf Backofen, Hans-Ulrich Krieger, Stephen P. Spackman,**Hans Uszkoreit (Eds.)*

Report of the EAGLES Workshop on Implemented Formalisms at DFKI, Saarbrücken
110 pages

D-93-26*Frank Peters*

Unterstützung des Experten bei der Formalisierung von Textwissen INFOCOM - Eine interaktive Formalisierungskomponente
58 Seiten

D-93-25*Hans-Jürgen Bürckert, Werner Nutt (Eds.)*

Modeling Epistemic Propositions
118 pages

Note: This document is available for a nominal charge of 25 DM (or 15 US-\$).

D-93-24*Brigitte Krenn, Martin Volk*

DiTo-Datenbank: Datendokumentation zu Funktionsverbgefügen und Relativsätzen
66 Seiten

D-93-22*Andreas Abecker*

Implementierung graphischer Benutzungsoberflächen mit Tcl/Tk und Common Lisp
44 Seiten

Note: This document is no longer available in printed form.

D-93-21*Dennis Drollinger*

Intelligentes Backtracking in Inferenzsystemen am Beispiel Terminologischer Logiken
53 Seiten

D-93-20*Bernhard Herbig*

Eine homogene Implementierungsebene für einen hybriden Wissensrepräsentationsformalismus
97 Seiten

D-93-16*Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Gabriele Schmidt*

Design & KI
74 Seiten

D-93-15*Robert Laux*

Untersuchung maschineller Lernverfahren und heuristischer Methoden im Hinblick auf deren Kombination zur Unterstützung eines Chart-Parsers
86 Seiten

D-93-14*Manfred Meyer (Ed.)*

Constraint Processing - Proceedings of the International Workshop at CSAM'93, St.Petersburg, July 20-21, 1993
264 pages

Note: This document is available for a nominal charge of 25 DM (or 15 US-\$).

D-93-12*Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein*

RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

D-93-11*Knut Hinkelmann, Armin Laux (Eds.)*

DFKI Workshop on Knowledge Representation Techniques - Proceedings
88 pages

Note: This document is no longer available in printed form.

D-93-10*Elizabeth Hinkelman, Markus Vonerden, Christoph Jung*

Natural Language Software Registry (Second Edition)
174 pages

D-93-09*Hans-Ulrich Krieger, Ulrich Schäfer*

*TDLE*ExtraLight User's Guide
35 pages

D-93-08*Thomas Kieninger, Rainer Hoch*

Ein Generator mit Anfragesystem für strukturierte
Wörterbücher zur Unterstützung von Texterkennung
und Textanalyse
125 Seiten

D-93-07*Klaus-Peter Gores, Rainer Bleisinger*

Ein erwartungsgesteuerter Koordinator zur partiellen
Textanalyse
53 Seiten

D-93-06*Jürgen Müller (Hrsg.)*

Beiträge zum Gründungsworkshop der Fachgruppe Ver-
teilte Künstliche Intelligenz, Saarbrücken, 29. - 30. April
1993
235 Seiten

Note: This document is available for a nominal charge
of 25 DM (or 15 US-\$).

D-93-05*Elisabeth André, Winfried Graf, Jochen Heinsohn,
Bernhard Nebel,**Hans-Jürgen Profitlich, Thomas Rist, Wolfgang Wahl-
ster*

PPP: Personalized Plan-Based Presenter
70 pages

D-93-04*Technical Staff*

DFKI Wissenschaftlich-Technischer Jahresbericht
1992
194 Seiten

D-93-03*Stephan Busemann, Karin Harbusch (Eds.)*

DFKI Workshop on Natural Language Systems: Reu-
sability and Modularity - Proceedings
74 pages

D-93-02*Gabriele Schmidt, Frank Peters, Gernod Laufkötter*

User Manual of COKAM+
23 pages

D-93-01*Philipp Hanschke, Thom Frühwirth*

Terminological Reasoning with Constraint Handling
Rules
12 pages

PROXIMA:

Ein System zur Generierung graphischer Abstraktionen

D-95-09

Document

Antonio Krüger