

Synchrones Generieren von Modellen in realen und virtuellen Räumen

Diplomarbeit

Vorgelegt von
Eckhard Meier

Bremen, 25. September 1998

Gutachter:
Prof. Dr. F. W. Bruns,
Prof. Dr. K.-H. Rödiger

Forschungszentrum Arbeit und Technik (artec)
Universität Bremen

Inhaltsverzeichnis

Kapitel 1 Einleitung und Zielsetzung	4
1.1 Motivation.....	7
1.2 Einordnung.....	11
Kapitel 2 Interaktionskonzepte	15
2.1 Direct Manipulation Interfaces.....	16
2.2 Virtual Reality Interfaces.....	19
2.3 Real Reality Interfaces	24
2.4 Alternativansätze und Projekte.....	28
2.4.1 Greifbare Benutzungsschnittstellen	29
2.4.1.1 Graspable User Interface	30
2.4.1.2 Natural User Interface	31
2.4.2 Sensored Objects	32
2.4.2.1 AlgoBlock.....	33
2.4.2.2 LEGOsheets.....	33
Kapitel 3 Synchrones Modellieren	36
3.1 Die Grundidee.....	36
3.2 Generelle Anforderungen.....	37
3.3 Analyse und Eingrenzung	39
3.3.1 Objektstrukturen.....	39
3.3.1.1 Bausteintypen	39
3.3.1.2 Bausteine	42
3.3.1.3 Modelle.....	43
3.3.2 Dynamik des Konstruktionsprozesses.....	44
3.3.2.1 Greifen.....	45
3.3.2.2 Bewegen	47
3.3.2.3 Loslassen	48
Kapitel 4 Spezifikationen	51
4.1 Datenstrukturen	51
4.1.1 Baukastensysteme.....	52
4.1.2 Elementtypen.....	53
4.1.3 Anknüpfungspunkte.....	54
4.1.4 Die Szene	57
4.1.5 Modelle	58

4.1.6	Elemente	59
4.1.7	Verbindungszustände	60
4.2	Algorithmen	61
4.2.1	Vorhersage der Bewegungsrichtung	61
4.2.2	Greifen und Zerlegen eines Modells	63
4.2.3	Bewegen eines Modells	65
4.2.3.1	Aktive Selektion	66
4.2.3.2	Vorselektion	67
4.2.3.3	Sekundärselektion	67
4.2.3.4	Primärselektion	68
4.2.4	Loslassen eines Modells	69
4.2.4.1	Kombinieren von Modellen	70
4.2.4.2	Absetzen eines Modells	70
Kapitel 5 Geschichte einer Implementierung		72
5.1	Entwicklungsrahmen	72
5.2	Technische Gerätschaft	73
5.3	Beobachtungen und Probleme	74
5.4	Selbsteinschätzung	76
5.5	Perspektiven	77
Literaturverzeichnis		82

Kapitel 1

Einleitung und Zielsetzung

Mit der fortschreitenden Entwicklung der Computertechnik haben sich im Laufe der Zeit auch die Umgangsformen mit Computern geändert. Die heute am weitesten verbreitete Form der Benutzungsschnittstelle zwischen Anwendern und Computern sind die sogenannten WIMP Systeme (windows, icons, mouse and pull-down menus) ([Shn1992], S. 207). Diese basieren auf der Eingabe von Daten mit Hilfe von Maus und Tastatur. Die Repräsentation von Informationen und Funktionalitäten erfolgt durch graphische Darstellungen, Text und Symbole (Icons) auf dem Bildschirm. Oftmals kommen bei solchen Benutzungsschnittstellen Metaphern zum Einsatz, die Analogien zwischen rechnerinternen Strukturen und den Benutzenden bekannte Umgebungen herstellen. Auf diese Weise soll ein besseres Verständnis des Umgangs mit dem Computer erreicht werden, da ein Bezug der Zustände des Systems auf eine reale Situation hergestellt werden kann.

Dieses Schnittstellendesign, auch als Direct Manipulation Interface bekannt, verspricht aufgrund der Möglichkeit, Informationen visuell aufbereiten und manipulieren zu können, eine einfache und intuitive Form der Benutzung. Es hat sich jedoch in der Praxis gezeigt, daß solche Systeme die an sie gestellten Erwartungen nicht oder nur teilweise erfüllen können. Wesentliche Einbußen zeigen sich hinsichtlich einer anschaulichen, leicht erkennbaren Darstellung komplexer Zusammenhänge, Strukturen und Funktionalitäten. Dadurch wird der intuitive Umgang mit einem System erschwert und läßt dieses unhandlich erscheinen, da die Erwartungen der Benutzenden nicht erfüllt werden:

„Such methods require all but the most expert user to stop thinking about the data, and to instead think about how to use the interface to manipulate the data.“
([Wex1993], S. 33)

Diese Problematik verschärft sich, wenn nicht nur zwei- sondern dreidimensionale Daten bearbeitet werden sollen. Sowohl die üblichen Ein- als auch Ausgabegeräte sind auf die Arbeit in maximal zwei Dimensionen ausgerichtet. Ihr Einsatz in einem dreidimensionalen Anwendungsgebiet bringt jedoch erhebliche Probleme mit sich, da nicht alle benötigten Freiheitsgrade zur Verfügung stehen:

„The most successful interface paradigm so far has been the Xerox Parc Desktop metaphor popularized among computer users by the Macintosh. However, if the desktop metaphor is well suited to interacting with two-dimensional worlds, it starts to show limitations when interacting with tree dimensional worlds.“
([Thal1991], S. 91)

In einem solchen Fall (zum Beispiel einer CAD-Anwendung) bleibt es im Wesentlichen der Vorstellungskraft der Benutzenden überlassen, zum einen gewünschte Manipulationen am dreidimensionalen Objekt vorab zu antizipieren und zum anderen die daraus resultierenden Ergebnisse von der zweiten in die dritte Dimension zu übertragen. Permanent muß ein gedanklicher Wechsel zwischen der zweiten und dritten Dimension vollzogen werden. Abbildungen 1 und 2 sind anschauliche Beispiele für eine unzureichende Darstellung dreidimensionaler Objekte in der Ebene, die keine eindeutige Interpretation zulassen.

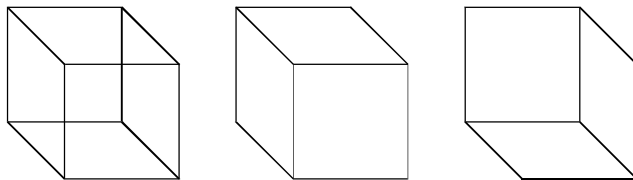


Abbildung 1: Die linke Darstellung des Würfels kann sowohl dem mittleren als auch dem rechten Standpunkt entsprechen.

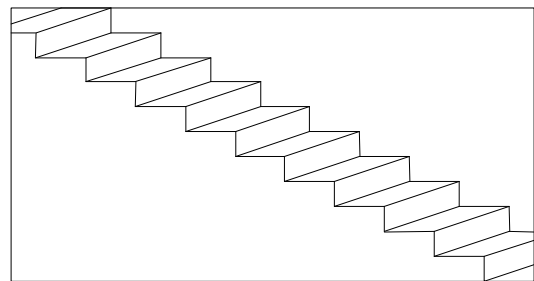


Abbildung 2: Die Treppe kann sowohl von oben als auch von unten betrachtet werden.

Die angeschnittenen Defizite graphischer Benutzungsschnittstellen und das Aufkommen neuer Ein- und Ausgabeperipherien haben zu neuen Ansätzen der Computerinteraktion geführt. Ihnen allen ist gemein, daß sie die Benutzungsschnittstelle um die dritte Dimension erweitern. Im Wesentlichen können folgende Bereiche mit unterschiedlichen Schwerpunkten genannt werden:

- **Virtual Reality:**
Benutzende tauchen gänzlich in eine künstliche, computergenerierte Welt ein.
- **Augmented Reality:**
Die Sicht der Realität wird durch Einblenden computergenerierter Bilder informell ergänzt.
- **Real Reality:**
Das reale Umfeld wird durch Sensorisierung der Hand als Interaktionsraum genutzt
- **Sensored Objects;**
Reale Objekte agieren mittels geeigneter Schnittstellen untereinander und/oder in ihrer direkten Umgebung.

Eine ausführlichere Darstellung der unterschiedlichen Konzepte erfolgt in Kapitel 2 der Arbeit.

Die hier vorgestellte Diplomarbeit entsteht im Rahmen des oben genannten Real Reality Benutzungskonzeptes. Dieses wurde 1993 im Forschungszentrum Arbeit und Technik der Universität Bremen von W. Bruns vorgestellt ([Bru1993a], [Bru1993b]) und fortan weiter entwickelt.

Kern dieses Interaktionskonzeptes ist die Integration bislang getrennter realer und virtueller Welten in einer Benutzungsschnittstelle. Dabei können auf Seiten der realen Welt die Vorteile des natürlichen, gewohnten und intuitiven Umgangs mit physikalischen Objekten genutzt werden, andererseits stehen durch den Computer auch die Möglichkeiten der computergetstützten Simulation als Mittel der rationalen und experimentellen Analyse und Verifikation von Modellen zur Verfügung (siehe [Bru1996a], S. 2).

Diese Diplomarbeit behandelt einen Teilaspekt aus diesem Anwendungsfeld des Modellierens im Realen und Virtuellen:

Den Schwerpunkt dieser Arbeit bildet die Spezifikation und Entwicklung einer Anwendung zur Montage komplexer statischer Modelle aus Bestandteilen eines Baukasten-Systems.

Die Konstruktion der Modelle erfolgt mit Hilfe realer Bausteine. Gleichzeitig wird durch die Aufzeichnung von Handbewegungen ein dreidimensionales Abbild der Originalkonstruktion in Form eines Computermodells generiert. Neben der Verwaltung einzelner Bausteine erfolgt zeitgleich eine Analyse der aktuellen Szenerie. Dabei wird die Lage einzelner Bausteine im Raum so interpretiert, daß diese entsprechend ihrer Struktur gruppiert und zu zusammengehörigen Einheiten verbunden werden. Ziel ist es, das für diesen Anwendungszweck wesentliche Verhalten der realen Gegenstände auf virtuelle Repräsentationen zu übertragen, so daß eine weitgehende Übereinstimmung der realen und virtuellen Szene gewährleistet ist.

Den Kern dieser Aufgabenstellung bildet die Entwicklung geeigneter Algorithmen hinsichtlich einer topologischen Szenenanalyse, sowie die Definition der für eine solche Analyse notwendigen Strukturen. Letztere stellen eine funktionale Erweiterung bereits vorhandener Objektgeometrien dar, die physikalische Eigenschaften real existierender Bausteine hinsichtlich ihrer Kombinationsmöglichkeiten abbilden. Aufgrund dieser Ergänzungen wird eine Möglichkeit geschaffen, komplexe Strukturen aus realen Bausteinen zu modellieren und in einem virtuellen Modell adäquat abzubilden.

Einen Themenbereich dieser Arbeit stellt dementsprechend die Spezifikation eines geeigneten Konzeptes und dessen Realisierung in Form einer prototypischen Anwendung dar. Eine wichtige Anforderung an die zu entwickelnde Konzeption stellt deren Generalisierbarkeit dar. Gefordert wird eine flexible Repräsentation von Verbindungseigenschaften montierbarer Objekte, um eine möglichst große Vielfalt einfacher Grundbausteine abbilden zu können.

Die Entwicklung der Algorithmen erfolgt in Anlehnung an bekannte Methoden der dreidimensionalen Bilderzeugung, Animation und Simulation. Die Struktur komplexer Modelle entspricht der eines ungerichteten Graphen, so daß hier auf entsprechende Literatur zurückgegriffen werden kann (Siehe Kapitel 4.2.4.1)

Weiterhin sollen in Erwägung gezogene Alternativansätze vorgestellt und diskutiert werden. Neben der Erläuterung konzeptioneller Ansätze erfolgt eine Darstellung der technischen Entwicklung und der damit einhergehenden Problemstellungen, Lösungsansätze und Designentscheidungen.

Im weiteren Verlauf dieses Kapitels werden die grundlegenden Prinzipien der Anwendung dargestellt und zu einer in sich geschlossenen Gesamtkonzeption zusammengefaßt. Abschließend sollen einige mögliche Anwendungsszenarien skizziert werden.

In Kapitel 2 werden die hier bereits angerissenen neuen Interaktionsformen dargestellt und deren wesentliche Unterscheidungsmerkmale in der Konzeption aufgezeigt. Zusätzlich werden einige mit dieser Arbeit verbundene Kernfragen anhand der vorgestellten Ansätze diskutiert.

Im darauffolgenden Kapitel 3 werden die Anforderungen an das zu entwickelnde Konzept erarbeitet, gefolgt von einer Analyse der Problemstruktur.

Nach dieser Darstellung wird in Kapitel 4 ein diesem Problem entsprechender Lösungsansatz diskutiert und in Form einer ersten Programmspezifikation konkretisiert.

In dem abschließenden Kapitel 5 erfolgt eine persönliche Darstellung der Prototypenentwicklung und ein kurzer Ausblick auf mögliche Erweiterungen des Systems und denkbare Anwendungsszenarien.

1.1 Motivation

Im Bereich der Computergrafik ist seit einiger Zeit vermehrt der Trend zur dreidimensionalen Visualisierung und Interaktion festzustellen. CAD-Programme unterstützen bereits seit längerem das direkte dreidimensionale Konstruieren am Bildschirm. Spezielle 3D-Konstruktionsprogramme, wie z.B. 3D-Studio MAX ([Ktx@]) oder Caligari Truespace ([Cal@]) ermöglichen aufwendige, realitätsnahe Präsentations- und Animationsgestaltungen am Computer. Im Umfeld des World Wide Web hat sich VRML (Virtual Reality Modeling Language) ([Ame1996], [Vrc@]) als Standard zur Beschreibung virtueller Welten etabliert. Im Bereich der Computerspiele hat die 3D-Darstellung bereits frühzeitig einen festen Platz eingenommen. Aufgrund der in diesem Unterhaltungssektor üblichen Kurzlebigkeit der Produkte und der stetig steigenden Anforderungen der Kunden, ist die technische Entwicklung in diesem Bereich weit fortgeschritten. Voll animierte virtuelle Szenarien, in denen sich mehrere Spieler gleichzeitig aufhalten können, gehören heute bereits zum Standard.

Aus meiner Sicht handelt es sich bei dieser generellen Entwicklung nicht um rein technische Spielereien, sondern um eine sinnvolle und logische Weiterentwicklung der Computertechnologie in Hinblick auf die dem Menschen angeborenen und trainierten Wahrnehmungs- und Handlungsgewohnheiten. Es wird prinzipiell die Möglichkeit geschaffen, eine dem Menschen zugänglichere Form des „Umgangs mit Computer“¹ zu ermöglichen.

Persönliche Erfahrungen mit 3D-Konstruktionsprogrammen im Rahmen des HYSIM-Projektes² ([Hys1997], [HYS@]) haben mir jedoch gezeigt, daß eine reine dreidimensionale Darstellung am Bildschirm für den Anwendenden zunächst keine neuen Handlungsspielräume eröffnet, sondern - ganz im Gegenteil - neue Probleme aufwirft.

Im Rahmen der prototypischen Entwicklung einer 3D- Benutzeroberfläche wurde von mir eine umfangreichere Szene (siehe Abbildung 3) modelliert. Während dieser Arbeit habe ich festgestellt, daß sich der Umgang mit der Software als ausgesprochen unhandlich und zeitaufwendig gestaltet. Die Gründe dafür sind in einer komplexen und abstrakten Funktionalität und prinzipiellen Einschränkungen der Benutzungsschnittstelle zu suchen:

- Aufgrund der zweidimensionalen Darstellung einer Szene am Bildschirm ergeben sich informelle Defizite (siehe Abbildung 4). Diese sollen durch die Anzeige mehrerer kleiner Ansichten gleichzeitig kompensiert werden:

„One long-standing difficulty in depicting spatial relationships is that most display devices are 2D. Therefore, 3D objects must be projected into 2D, with considerable attendant loss of information – which can sometimes create ambiguities in the image.“ ([Fol1990], S.608)



Abbildung 3: Entwurf eines dreidimensionalen Büroraumes.

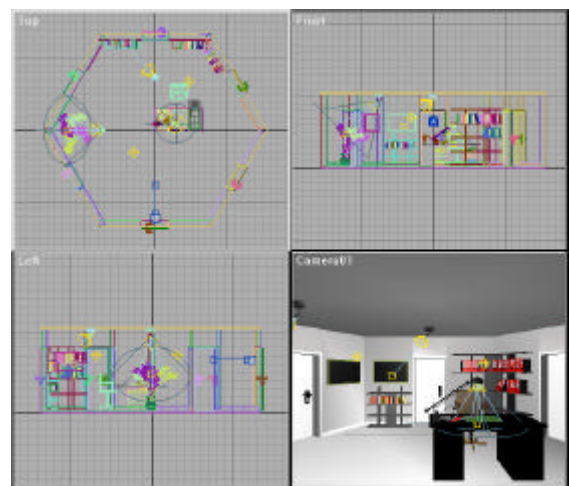


Abbildung 4: Ausschnitt der Benutzungsschnittstelle einer 3D-Gestaltungsanwendung.

¹ Viel zitierter Ausspruch Frieder Nakes. Er bringt zum Ausdruck, daß „der Computer an sich“ als Mittel zum (Anwendungs-) Zweck, d.h. als Medium zu betrachten ist. Erst durch die Art der Anwendung kann eine geeignete Form des Umgangs mit dem Computer als Medium gestaltet werden.

² Hypermediagestützte Simulationssysteme für berufliche Schulen.

Benutzende müssen durch die Kombination der einzelnen Ansichten eine gedankliche Rekonstruktion der Szene vornehmen, um sich die genaue Gestalt der dargestellten Objekte vor Augen führen zu können. Dabei kommt erschwerend hinzu, daß die Darstellung am Bildschirm aus Gründen mangelnder Rechenkapazitäten oft nur als Drahtgittermodell erfolgt, so daß bereits der rein optische Eindruck starken Einschränkungen unterliegt.

- Die Maus, als ein auf zweidimensionales Interagieren spezialisiertes Eingabegerät, ist aufgrund des fehlenden dritten Freiheitsgrades nur eingeschränkt für räumliches Arbeiten geeignet. Dieser Mangel wird von der Software kompensiert, indem nur in Einzelschritte unterteilte oder auf eine Ebene beschränkte Interaktionen zugelassen werden. Dieser Mangel an Manipulationsfreiräumen führt (zumindest bei einem ungeübten Benutzer) zu einer stockenden, langwierigen und fehlerbehafteten Arbeitsweise.
- Die Anwendungen sind sehr stark an internen Datenstrukturen ausgerichtet. In der Regel werden Flächenmodelle erzeugt, die sich aus einer Anzahl von Polygonen zusammensetzen. Einzelne Objekte bestehen lediglich aus einer Hülle ohne Substanz¹. Die von der Anwendung bereitgestellten Manipulationswerkzeuge bieten eine umfangreiche Funktionalität, die auf die Bearbeitung speziell dieser Flächenmodelle ausgerichtet ist. Anwendende müssen sich demzufolge nicht mit Abbildungen ihrer dinglichen Umwelt auseinandersetzen, sondern werden mit abstrakten (Daten)Strukturen konfrontiert, die mit dem tatsächlichen Gegenstand wenig gemeinsam haben. Die Umsetzung eines Vorhabens in geeignete Arbeitsschritte gestaltet sich entsprechend schwierig.
- Animationen erfordern die Handhabung einer weiteren Dimension. Auch für dieses Arbeitsfeld fehlen geeignete Interaktionsmittel. Die Bewegung einer Maus kann entweder als zeitliche oder örtliche Veränderung interpretiert werden ([Fit1995], S. 2) Zeitliche Abläufe müssen stufenweise definiert und von den Anwendern mit Hilfe aufwendiger Verfahren, wie z.B. Key-Frame Animation, erstellt und verifiziert werden. Aus meiner Sicht ist spätestens hier eine Grenze erreicht, an der die Vorstellungskraft der Benutzenden die Mängel einer reinen Bildschirmdarstellung nicht mehr kompensieren kann.

Anhand dieser Probleme zeigt sich, daß die Möglichkeiten der dreidimensionalen Darstellung zwar einen informellen Gewinn gegenüber herkömmlichen Repräsentationsformen mit sich bringen können, grafische Benutzungsschnittstellen bieten jedoch keine geeigneten Mittel, um mit solchen Daten in einer angemessenen Art und Weise umgehen zu können.

¹ Die genau entgegengesetzte Form des Flächenmodells ist das sogenannte Volumenmodell (CSG - Constructive Solid Geometry). Dabei wird ein Objekt aus mathematisch definierten Grundkörpern zusammengesetzt, die mit Hilfe boolescher Operationen kombiniert werden. Die grundlegende Problematik der anschaulichen Repräsentation bleibt jedoch auch hier bestehen.

Dieser Problematik widmen sich die bereits genannten Interaktionskonzepte. Deren wesentliche Zielsetzung ist es, das computergestützte Arbeiten in einem dreidimensionalen realen oder virtuellen Umfeld so zu gestalten, daß der Umgang mit derartigen Systemen eine dem gewohnten menschlichen Handeln entsprechende Form annimmt.

Im Rahmen des Real Reality User Interfaces wird ein Ansatz verfolgt, dem die Integration realer und virtueller Objekte in eine Benutzungsschnittstelle als zentrale Grundidee zugrunde liegt. Der Gebrauch physikalischer Modelle bietet einen sinnlichen, intuitiven, anschaulichen und leicht verständlichen Zugang zu einer Sachlage oder Problemstellung, während sich die computergestützte Simulation auszeichnet für strukturelle, physikalische und/oder hypothetische Betrachtungen einsetzen läßt. Die synchrone Manipulation realer Gegenstände und entsprechender virtueller Abbilder ermöglicht es, die Vorteile der jeweiligen Bereiche zu kombinieren und den Anwendenden gleichzeitig zwei unterschiedliche Perspektiven derselben Grundsituation bereitzustellen.

Gerade hinsichtlich des Entwurfs einzelner, statischer Modelle erscheint diese neue Form der Mensch-Maschine-Kommunikation vielversprechend. Die Möglichkeiten, direkt an real existierenden Konstruktionen gestalten und experimentieren zu können und gleichzeitig ein adäquates virtuelles Abbild zur Verfügung zu haben, bringen diverse Vorteile gegenüber einer rein rechnergestützten Vorgehensweise mit sich.

So wirken sich zum Beispiel physikalische Gesetzmäßigkeiten direkt und unmittelbar auf ein gegenständliches Modell aus. Bei der Gestaltung eines Raumes stehen die darin platzierten Möbel selbstverständlich auf dem Fußboden. Dies ist bei der Arbeit am Computer nicht so. Es ist durchaus üblich, daß Objekte frei im Raum schweben und von den Benutzenden erst explizit auf dem Boden, der natürlich auch erst erstellt werden muß, platziert werden müssen.

Ein weiterer entscheidender Vorteil eines physikalischen Modells gegenüber einer zweidimensionalen Darstellung ist dessen Plastizität. In der Regel erkennen Modellierende und Unbeteiligte unmittelbar wesentliche Merkmale eines Gebildes, ohne daß genauere Analysen oder Beschreibungen nötig sind. Das Modell ist selbsterklärend, da es einen ganzheitlich räumlichen Eindruck vermittelt, dessen Qualität auf der Bildschirmfläche nicht erreicht werden kann.

Diese Vorteile des Real Reality Konzeptes reduzieren den Grad der Abstraktion im Vergleich zu einer herkömmlichen Benutzungsschnittstelle erheblich. Die Frage „Was will ich konstruieren?“ tritt gegenüber der Frage „Wie muß ich konstruieren?“ stark in den Vordergrund und schafft Freiräume für eine ungestörte kreative Arbeitsweise.

1.2 Einordnung

Im Rahmen des RUGAMS¹- und EUGABE-Projektes² ([RUG@], [EUG@]) des Forschungszentrums Arbeit und Technik wurden bereits erste Anwendungen des Real Reality Konzeptes entwickelt, die zum Beispiel den Planungsprozeß förder technischer Anlagen anhand eines gegenständlichen Modells unterstützen ([Bra1997], [Schä1997]).

Die virtuelle Repräsentation der benutzten realen Objekte beschränkt sich dabei zunächst auf rein geometrische Eigenschaften. Funktionale Merkmale der Dinge werden nicht in den Prozeß des Modellierens eingebunden und können aus diesem Grunde nicht vom System berücksichtigt werden ([Rob1997]). Demzufolge lassen sich existierende Beziehungen zwischen realen Körpern, die auf deren physikalischen Eigenschaften beruhen, nicht auf das virtuelle Modell übertragen. Gegenstände, die von den Benutzenden an einem bestimmten Ort platziert wurden, können ihre Position und Lage nicht selbständig ändern, sondern werden während der Modellierung ausschließlich durch die „magische Hand“ der Benutzenden bewegt. Auch kollidieren sie nicht miteinander, fallen nirgends herunter, brechen nicht auseinander, werden nicht heiß usw. Im Gegensatz zu ihren realen Pendanten unterliegen die virtuellen Objekte keinerlei physikalischen oder anderen Gesetzmäßigkeiten.

Betrachtet man das Modellieren als einen in sich geschlossenen Prozeß, so gibt es darin keine gegenständliche Komponente mit einem eigendynamischen Verhalten. Die Reduktion der gegenständlichen Objekte auf ihr Erscheinungsbild stellt deshalb zunächst eine grundsätzliche Einschränkung der generierbaren Modelle und damit des Modellierprozesses dar. Abgebildet werden nur solche Konstruktionen, für die existierende physikalische Abhängigkeiten zwischen Körpern irrelevant sind. Die Topologie innerhalb eines Modells wird ausschließlich und unmittelbar von den Benutzenden bestimmt und beeinflußt.

Anwendungen dieser Art zeichnen sich in der Regel dadurch aus, daß, basierend auf einer vornehmlich statischen Anordnung von Gegenständen, eine funktionale Ebene aufsetzt. Diese verwaltet funktionale Zusammenhänge, wie sie zum Beispiel für eine computergestützte Simulation benötigt werden, ist aber von dem geometrischen Basismodell weitgehend entkoppelt. Die funktionalen Informationen werden lediglich auf die Geometrien einer vorab erstellten Szene projiziert, ohne daß es sich dabei um physikalische Charakteristika der Objekte handeln muß. Die Trennung der objektbezogenen Modellschicht von der funktionsorientierten Simulationsschicht wird in Abbildung 6 deutlich.

¹ Rechnergestützte Übergänge zwischen gegenständlichen und abstrakten Modellen produktionstechnischer Systeme.

² Erfahrungsorientierte Übergänge zwischen gegenständlichen und abstrakten Modellen für die berufliche Qualifikation.

Am Beispiel eines realisierten, auf gegenständlichem Vormachen basierendem Teach-In Verfahren von Verzweigungsregeln läßt sich diese Trennung darlegen. Mit Hilfe einfacher Bauklötze wird es den Anwendenden ermöglicht, Regeln einer Materialflußsimulation zu bestimmen (siehe Abbildung 5). Dazu werden die Bewegungsbahnen der gegriffenen Bausteine aufgezeichnet und in Zuteilungsregeln umgesetzt. Unterschiedliche Farben der Objekte dienen als Unterscheidungsmerkmal unterschiedlicher Transportwege ([Bra1997], S. 4).

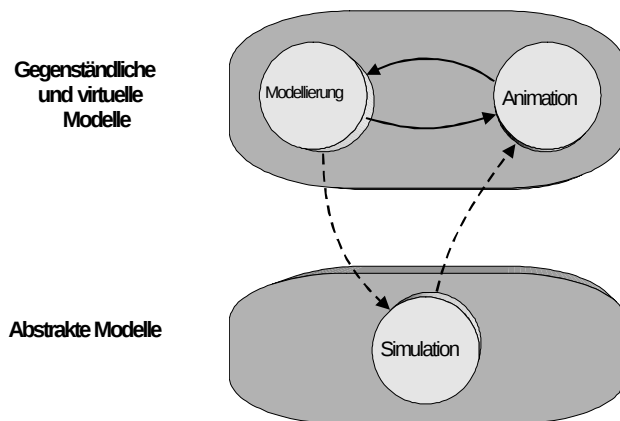


Abbildung 6: Beziehungen zwischen Modellierung, Simulation und Animation.



Abbildung 5: Gegenständliches Vormachen einer Verzweigungsregel.

In diesem konkreten Beispiel definiert die Bewegung eines Bausteins eine für die Simulationsschicht relevante Funktionalität. Dabei spielen die physikalischen Eigenschaften des Bausteins zunächst eine untergeordnete Rolle. Für die reine Interpretation des Transportweges ist es nicht relevant, ob das Objekt rund oder quadratisch, groß oder klein, leicht oder schwer ist. Es muß lediglich ein Unterscheidungsmerkmal für die Bestimmung unterschiedlicher Wege definiert werden.¹ Dem Objekt kommt im Rahmen der Simulationsbeschreibung eine eher symbolische Bedeutung zu, denn es repräsentiert eine Klasse zu transportierender Gegenstände.

Der von mir entwickelte Ansatz unterscheidet sich von dem hier vorgestellten insofern, daß bereits im Rahmen der Modellgestaltung eine Erweiterung der für ein Objekt charakteristischen Eigenschaften um funktionale Aspekte erfolgt. Im Speziellen wird den virtuellen Bausteinen eines Baukastensystems, analog zu ihren realen Repräsentanten, die Fähigkeit verliehen, Verbindungen mit anderen einzugehen und wieder zu lösen. Auf diese Weise wird die isolierte Behandlung der Gegenstände *während* des eigentlichen Modellervorgangs auf-

¹ Ich beziehe mich hier ausdrücklich auf die Definition reiner Funktionalität, die im wesentlichen durch das virtuelle Objekt erfolgt. Im Hinblick auf die Intention des Real Reality Konzeptes kommt dem gewählten realen Objekt hingegen eine umfassendere Bedeutung zu. Dessen physikalische Eigenschaften können für die Benutzenden sehr wohl von Bedeutung sein, da durch den gegriffenen Baustein wiederum ein weiterer, eben der zu transportierende Gegenstand bezeichnet wird. Schwere Bausteine können zum Beispiel volle und damit schwere Paletten symbolisieren, wo hingegen leichte Bausteine mit leeren Paletten assoziiert werden können.

gehoben. Zusammengesteckte Grundelemente werden zu einer logischen Einheit, dem Modell, zusammengefaßt. Veränderungen der Lage eines gegriffenen Elements durch die Benutzenden wirken sich aufgrund der gegebenen Objekteigenschaften auf die Topologie der gesamten Szene aus.

Genauer gesagt können Verbindungen zwischen Objekten erstellt und wieder aufgehoben werden. Zusätzlich müssen untereinander verbundene Steine ihre Lage und Position entsprechend ihrer Partner anpassen. Das Verhalten der einzelnen virtuellen Objekttypen begründet sich dabei auf die ihnen zugrundegelegten funktionalen Eigenschaften ihrer physikalischen Vorbilder.

Der Einsatz einer solchen Anwendung wird in der Regel auf die Modellierung einzelner, komplex strukturierter Konstruktionen ausgerichtet sein. Im Mittelpunkt steht dabei der Aspekt des technisch orientierten Gestaltens und Experimentierens mit gegenständlichen Grundelementen. Die daraus resultierenden Ergebnisse lassen sich als Basis für weitere Anwendungsfelder, zum Beispiel für Simulationszwecke, weiter verwenden und ergänzen. Diese sehr interessante Möglichkeit wurde bereits im Rahmen der Real Reality Konzeption als vielversprechende Perspektive erkannt:

„Einsatz im Experiment

Mit einem Mechanikbaukasten, der aus verschiedenen Grundbausteinen besteht, können materielle Modelle gebaut werden, deren immaterielles Abbild gleichzeitig im Rechner entsteht. Das Abbild kann auch über die Demontage des realen Objektes hinaus erhalten und variiert werden. Der kreative erstmalige Bauvorgang findet im stofflichen Bereich statt, während die Analyse, Variation, Duplikation und Kombination zu größeren Systemen mit Hilfe des Rechners erfolgen kann. Dadurch sind Übergänge und Bezüge zwischen einer konkreten, stofflichen Welt der Einzelfälle und der abstrakten, virtuellen Welt der Gesetzmäßigkeiten möglich.“ ([Bru1996b], S. 7)

Der wesentliche Unterschied zwischen den hier vorgestellten Ansätzen besteht in der Zuordnung der in einem Modell erfaßten Funktionalität. In dem vom RUGAMS-Projekt verfolgten Konzept wird eine globale Sicht einer statischen Grundsituation eingenommen. Die zu realisierende Funktionalität wird nachträglich in eine vorhandenen Szene eingefügt¹, d.h. sie ist von dem Prozeß des Modellaufbaus weitestgehend entkoppelt. Die Gültigkeit einer Szene, die mit Hilfe dieses Ansatzes modelliert wurde, bleibt auch dann noch erhalten, wenn die funktionalen Zusatzinformationen von den Grundobjekten entfernt werden. Das Modell ließe sich auch in diesem Falle noch uneingeschränkt manipulieren, sofern nicht auf die Ebene der Simulation zurückgegriffen wird.

Im Gegensatz dazu wird in dem hier vorgestellten Ansatz ein Typ von Bausteinen um funktionale Charakteristika erweitert. Diese sind integraler Bestandteil eines Objektes und

¹ Die Unterscheidung der zwei Ebenen wird anhand der Software deutlich, da explizit zwischen einem Modellierungs- und einem Teach-In-Modus gewechselt werden muß.

somit fest in den Modellierprozeß eingebunden. Das Herauslösen der Verbindungseigenschaften aus einer solchen Konstruktion würde dessen Brauchbarkeit stark einschränken, da die logische Struktur des Modells verloren ginge. Eine weitere Bearbeitung der Szene wäre nur eingeschränkt möglich und in der Regel nicht sinnvoll.

Die Abweichungen der dargelegten Ansätze verdeutlichen auch die mit ihnen verbundenen unterschiedlichen Zielsetzungen. Während sich der erste aufgrund seiner *globalen* Erfassung von Topologien für die Abbildung systemumfassender Simulationen, wie zum Beispiel der Layoutgestaltung von Produktionsanlagen, eignet, konzentriert sich der zweite Ansatz stärker auf die Eigenschaften *einzelner* Objekte und deren Kombinationsmöglichkeiten zwecks Konstruktion komplexer Gebilde, beispielsweise von Förderband-Modellen.

So zeigt sich, daß es sich bei dieser Anwendung nicht um ein konkurrierendes System handelt, sondern um eine Erweiterung bisher realisierter Anwendungen im Sinne des Real Reality Ansatzes.

Kapitel 2

Interaktionskonzepte

Interaktionskonzepte bestimmen als grundlegender Bestandteil von Benutzungsschnittstellen die Form der Mensch-Maschine-Kommunikation. Auf der Grundlage gegebener technischer Hilfsmittel wird festgelegt, auf welche Art und Weise mit einem Computer kommuniziert werden kann. Die Gestalt der Ein- und Ausgabetechniken ist auch für den Grad der Abstraktion zwischen den Vorstellungen der Benutzenden und den vom System akzeptierten bzw. repräsentierten Informationen ausschlaggebend. So müssen im Falle einer textbasierten Schnittstelle zum Beispiel alle Vorhaben der Anwendenden in eine dem System bekannte Syntax übersetzt werden. Demgegenüber erlauben grafische Schnittstellen ein auf Gesten (point and klick, drag and drop Methoden) basierendes und damit in der Regel anschaulicheres Vorgehen. Die realisierten Interaktionstechniken prägen die Umgangsform mit einem System und tragen entscheidend zur Handhabbarkeit und Akzeptanz einer Anwendung bei.

Da sich diese Arbeit mit einem speziellen Aspekt der Computerinteraktion auseinandersetzt, sollen im weiteren Verlauf dieses Kapitels die Grundideen einiger relevanter Interaktionskonzepte dargestellt werden. Begonnen wird mit der Erläuterung des Direct Manipulation Interfaces, welches den aktuellen Stand in der Entwicklung der Computerinteraktion darstellt und als Basis der darauffolgend erläuterten vier Entwicklungsbereiche angesehen werden kann.

2.1 Direct Manipulation Interfaces

Direct Manipulation Interfaces finden primär in den heutzutage eingesetzten Computersystemen Verwendung. Die Mehrheit der Graphical User Interfaces (GUI) greifen auf diese Form der Interaktion zurück. Die Möglichkeiten der visuellen Darstellung von Informationen am Bildschirm und die Fähigkeit, diese graphischen Repräsentationen visuell manipulieren zu können, stellen die entscheidende Grundlage dieser Benutzungsschnittstellen dar:

“A direct-manipulation user interface is one in which the objects, attributes, or relations that can be operated on are represented visually; operations are invoked by actions performed on the visual representations, typically using a mouse. That is, commands are not invoked explicitly by such traditional means as menu selection or keyboarding; rather, the command is implicit in the action on the visual representation.” ([Fol1990], S. 397)

Primäres Ziel dieser direkten Interaktionsform ist es, eine einfache, intuitive und unkomplizierte Handhabung des Computers zu gewährleisten. Fragestellungen des angestrebten Arbeitsziels sollen gegenüber Problemstellungen der Benutzungsführung möglichst stark in den Vordergrund gestellt werden, so daß sich Anwendende voll und ganz auf ihre inhaltliche Aufgabenstellungen konzentrieren können. Dazu gilt es, den von einer Anwendung geforderten kognitiven Aufwand weitestgehend zu minimieren, um dadurch den Eindruck der Direktheit zu verstärken ([Hut1986], S. 93., [Shn1992], S. 202).

Der graphischen Informationsaufbereitung kommt in diesem Zusammenhang eine besondere Bedeutung zu, da erst auf diesem Weg internen Datenstrukturen und Zusammenhängen ein anschaulicher Charakter verliehen werden kann. Die Darstellung eines Diagramms wird zum Beispiel gegenüber einer tabellarischen Auflistung von Einzelwerten eine bessere Interpretation der zugrunde gelegten Daten erlauben.

Indem interne Zustände eines Systems durch graphische Objekte symbolisiert werden, kann ein auf das Anwendungsfeld angepaßtes, konzeptionelles Modell erstellt werden. Dieses Modell dient als Kommunikationsmedium zwischen System und Benutzenden und ist als fester Bestandteil der Benutzungsschnittstelle aufzufassen.

Mit Hilfe einer solchen Modellwelt können die Anwendenden einen anschaulichen Zugang zu den abstrakten, im System verborgenen Strukturen erlangen, indem sie ihre Erfahrungen und Zielsetzungen in den Kontext dieser repräsentierten Welt projizieren. Das konzeptionelle Modell dient generell als Bezugsraum für Zustandsinterpretationen und darüber hinaus als Basis für die Planung weiterer Vorgehensweisen. Der Benutzungsschnittstelle kommt in diesem Gesamtkontext die Aufgabe zu, die von den Anwendenden ausgeführten Objektmanipulationen zu interpretieren und in geeignete Systemkommandos umzusetzen. Daraus resultierende Ergebnisse müssen sich in das Gesamtbild der Modellwelt einfügen und gleichzeitig den Intentionen der Benutzenden entsprechen.

Hutchins ([Hut1986]) unterscheidet in diesem Zusammenhang zwei unabhängige Bereiche, die von dem Interface abgedeckt werden müssen.

Hinsichtlich der Dateneingabe ist eine Kluft der Befehlsausführung („*Gulf of Execution*“) zu überbrücken. Diese Kluft bezeichnet den Unterschied zwischen der antizipierten Herangehensweise an eine Problemstellung und den physikalischen Anforderungen, die das System zwecks Umsetzung an die Anwendenden stellt. Umgekehrt beschreibt sie die augenscheinliche Differenz zwischen Informationsausgabe und den erwarteten Vorstellungen der Anwendenden. Um diese Kluft seitens der Schnittstelle möglichst weit zu schließen, müssen die eingesetzten Interaktionsmechanismen soweit wie möglich an den Überlegungen und Absichten der Benutzenden ausgerichtet werden.

Entsprechend der Informationseingabe existiert auf der Ausgabeseite eine Evaluierungskluft („*Gulf of Evaluation*“), die es ebenfalls zu überwinden gilt. Diese gibt die Differenz einer mentalen Anschauung der Benutzenden in Bezug auf die visualisierte Modellwelt wieder. Entspricht das Verhalten von Objekten nicht den Vorstellungen der Anwendenden, kommt es zu einem Widerspruch. Dieser muß aufgelöst werden, bevor eine korrekte Interpretation der Gesamtsituation möglich ist, d.h. die Benutzenden müssen eine Anpassung ihrer Denkweise an die Modellwelt des Systems vornehmen.

Um solche Beeinträchtigungen zu vermeiden, muß mit Hilfe der Benutzungsschnittstelle der Eindruck vermittelt werden, bei den symbolischen Repräsentationen handle es sich um tatsächliche, autonome Objekte, deren Eigenschaften und Verhalten mit den Vorstellungen der Anwendenden weitestgehend übereinstimmen. Die Schnittstelle sollte den Anschein des aktiven Agierens innerhalb der Modellwelt erwecken.

Die Autoren stellen fest, daß das Überbrücken der zwei Kluften durch das System die entscheidende Grundlage für ein Direct Manipulation Interface darstellt. Das Einbinden der Benutzenden in eine aufgabenbezogene Modellwelt reduziert die Inanspruchnahme kognitiver Leistungen und erzeugt so ein Gefühl der direkten Integration und Manipulation:

“Minimum processing is required to get from intention to action and from outcome to interpretation. ... The hope is that people will be able to spend their time learning the task domain, not learning the computer system. Indeed, a properly developed Direct Manipulation interface should appear to the user as if it is the task that is being executed directly: The computer system and its interface will be more or less invisible.” ([Hut1986], S. 123)

Einen ähnlichen Ansatz der direkten Manipulation verfolgt Brenda K. Laurel ([Lau1986]), indem sie einen Vergleich zwischen Benutzungsschnittstellen und Theateraufführungen zieht. Sie legt dar, daß Schnittstellen – wie auch Theaterstücke - eine in sich geschlossene, nachvollziehbare Welt aufbauen müssen, damit sich Anwendende rational und emotional auf diese einlassen können. Objekte dieser Welt besitzen einen nachahmenden („mimetic“) Charakter, d.h. sie lassen einen Bezug zur Realität erkennen und bilden zusammengefaßt ein in sich geschlossenes, konsistentes System. Der Kontext dieses Systems erlaubt Personen einen

„runden“, in sich schlüssigen Umgang mit einer Anwendung, sofern sie in den mimetischen Gesamtzusammenhang der (Modell-) Welt eingebunden werden.

Überdies spielt der von den Benutzenden eingenommene Standpunkt eine entscheidende Rolle. Direkte Manipulation setzt eine Perspektive der ersten Person („first-personness“) voraus, denn nur so wird ein Weg des direkten, eigenständigen Handelns geschaffen:

„The conclusion can only be that a particular kind of relationship between the user and the context is appropriate to the nature and form of an interactive mimesis. The term I have chosen to describe it is first-personness. The first-person metaphor is grammatical: The personness of pronouns reflects where one stands in relation to others and to the world. Most movies and novels, for example, are third-person experiences; the viewer or reader is “outside” the action, and would describe what goes on using third-person pronouns: ...“ ([Lau1986], S. 76)

Grundsätzliches Ziel eines solchen mimetischen Anwendungskonzepts ist es, eine ausgeprägte Interaktion aus der Perspektive der ersten Person zu realisieren, wobei Ein- und Ausgabemethoden an dem Vorbild der realen Welt zu gestalten sind. Die Benutzungsschnittstelle soll in letzter Konsequenz, u.a. durch den Einsatz multimodaler Technologien, den Eindruck des sich-dort-Aufhaltens und aktiven Handelns vermitteln.

Der Begriff des „Direct Manipulation Interface“ wurde wesentlich durch Ben Shneiderman ([Shn1992], S. 181 ff.) geprägt. Entsprechend der bisher vorgestellten Autoren hebt auch er die modellhafte Abbildung eines anwendungsbezogenen Realitätsausschnittes als einen entscheidenden Aspekt des Benutzungskonzeptes hervor. Folgende drei Prinzipien der direkten Manipulation werden von ihm definiert ([Shn1992], S. 205):

1. *Continuous representation of the objects and actions of interest*
2. *Physical actions or presses of labeled buttons instead of complex syntax*
3. *Rapid incremental reversible operations whose effect on the object of interest is immediately visible.*

Aus diesen Richtlinien leitet er sieben Eigenschaften direkt manipulativer Systeme ab, auf deren Darstellung an dieser Stelle verzichtet werden soll.

Die hier vorgestellten Konzepte der direkten Manipulation decken sich in einigen grundlegenden Annahmen und Forderungen bezüglich der Gestaltung von Benutzungsschnittstellen. Dabei steht die Fragestellung nach einer geeigneten Verzahnung der realitätsbezogenen Erfahrungs- und Vorstellungswelt der Benutzenden und der abgebildeten, künstlichen Modellwelt der Anwendungsschnittstelle im Vordergrund.¹ Die Konformität

¹ Oftmals werden Metaphern herangezogen, um eine Brücke zwischen realer und Computerwelt zu schlagen. Dabei werden computerinterne Strukturen in einer Form repräsentiert, die einen den Benutzern bekannten Ausschnitt der realen Welt darstellt. Die bekannteste dieser Metaphern - die Desktop-Metapher - repräsentiert Informationen im Kontext eines Büros unter Einsatz von Symbolen wie Ordnern, Papierkorb, Postfach usw.

dieser beiden Perspektiven ist entscheidend für die Handhabbarkeit einer Applikation. Auftretende Inkonsistenzen können nur durch einen zusätzlichen Interpretationsaufwand der Betroffenen kompensiert werden, in dessen Folge ein thematisch ausgerichteter, fließender Arbeitsprozeß zwangsläufig zerschlagen wird.

Die Praxis jedoch zeigt, daß diese angestrebte Form der Mensch-Maschine-Kommunikation derzeit nicht die erwarteten Arbeitserleichterung realisiert. Bereits der Umgang mit der Maus gestaltet sich alles andere als intuitiv und realitätsnah. Die genaue Positionierung des Mauszeigers und der sogenannte Doppelklick stellen einen ungeübten Benutzer vor ernsthafte Probleme. Zudem sind die im Rahmen der Modellwelt eingesetzten Metaphern in der Regel nicht konsistent einzuhalten, so daß sich hier zwangsläufig semantische Brüche ergeben, und auch der Einsatz abstrakter graphischer Symbole (Icons) reicht oftmals nicht aus, um die dahinter verborgene Funktionalität anschaulich und auf den ersten Blick verständlich zu visualisieren ([Shn1992], S. 204).

Aus meiner Sicht bleibt deshalb festzustellen, daß die heute üblichen Formen der Computer-interaktion nicht die notwendigen Mittel bereitstellen können, um eine einfache, intuitive und unproblematische Handhabung künstlicher Modellwelten zu ermöglichen.

Die im Weiteren dargestellten Konzepte verfolgen auf unterschiedlichen Wegen das wesentliche Ziel, gerade diese, nach wie vor erhebliche Diskrepanz zwischen realer und konzeptioneller Welt weiter zu reduzieren. Dazu wird in der Regel der Ansatz verfolgt, Parallelen zur realen vertrauten Umwelt herzustellen und in das System zu integrieren.

2.2 Virtual Reality Interfaces

Der Begriff „Virtual Reality“ (VR) oder auch „Virtual Environment“ (VE) bezeichnet eine Form der Mensch-Maschine-Kommunikation, die sich durch die Interaktion in simulierten, dreidimensionalen Welten auszeichnet. Diese visualisierten Umgebungen orientieren sich in ihrer Darstellung und ihrem Verhalten an der Realität ([Pso1995], S. 3), so daß ein Perspektivwechsel der Benutzenden weg von einer externen, betrachtenden - hin zu einer internen, partizipierenden Grundhaltung vollzogen werden kann ([Bri1990], S. 2). Durch den Eindruck der persönlichen Präsenz soll die Illusion einer in sich geschlossenen, eigenständigen Welt vermittelt und aufrecht erhalten werden.

Die Grundidee des Eintauchens (Immersion) in eine künstliche Umgebung ist in unterschiedlicher Ausprägung fester Bestandteil des VR-Ansatzes. Diese vollständige Integration der Benutzenden in die virtuelle Welt soll mit Hilfe einer möglichst umfassenden Einbindung der menschlichen Sinne durch die Benutzungsschnittstelle erzielt werden. Angestrebt wird eine natürliche Interaktionsform mit dem System und die Erweiterung der Wahrnehmungs- und Aktionsmöglichkeiten der Akteure. Das ausgeprägte Einbeziehen und Ansprechen der

menschlichen Wahrnehmung soll einen Zuwachs an Handlungsspielräumen schaffen, um somit das Sammeln direkter Erfahrungen zu ermöglichen ([Laur1997], S. 1, 2, 6, 16).

Folgende Kategorien unterschiedlich komplexer virtueller Welten mit variierenden Interaktionsformen werden generell unterschieden:

- Desktop oder „fishtank“ VR
Der Monitor dient als Sichtfenster in eine Virtuelle Welt. Er repräsentiert die aktuelle Blickrichtung der Benutzenden innerhalb der virtuellen Welt. Um den Eindruck der dritten Dimension stärker hervorzuheben, werden ggf. Stereo-Brillen eingesetzt.
- Immersive VR
Benutzende tauchen gänzlich in eine künstliche Welt ein. Ihnen wird mittels unterschiedlicher technischer Ein- und Ausgabemedien der Anschein von Gegenwärtigkeit vermittelt.
- Distributed VE
Systeme, die eine Simulation virtueller Welten auf mehrere, untereinander vernetzte Computer ermöglichen.
- Multi-User VE und Collaborative VE
Können mehrere Benutzer gleichzeitig an einer virtuellen Welt partizipieren, spricht man von Multi-User Virtual Environments. Können die Benutzenden darüber hinaus nicht nur innerhalb der Umgebung interagieren, sondern auch *miteinander* kooperieren, so handelt es sich um sogenannte Collaborative Virtual Environments.

Das dem Bereich Virtual Reality zugrundeliegende Konzept, Computer als Realitäts-Generatoren ([Bri1990], S. 1) einzusetzen, stellt eine konsequente Erweiterung des bereits dargestellten Direct Manipulation Ansatzes dar. Die Forderung nach einer konsistenten, an der realen Welt angelehnten Modellwelt wird unter Einsatz neuer Ein- und Ausgabemedien, und der Integration der Anwenden in eine geschlossene, synthetische Umgebung konsequent fortgesetzt.

Es wird der Versuch unternommen, eine subjektive Wahrnehmung der künstlichen Umgebung und somit ein Gefühl der persönlichen Anwesenheit mit entsprechenden Handlungsspielräumen zu erzielen. Das Computersystem übernimmt dabei die Funktion eines Kommunikations- und Interaktionsmediums. Es ist vermittelnde (und somit auch kontrollierende) Instanz zwischen einer synthetischen Welt und den „darin“ befindlichen Personen.

Um diesem formulierten Realitätsanspruch gerecht werden zu können, müssen hinsichtlich der Mensch-Maschine-Kommunikation vielschichtige und komplizierte Beziehungsnetzwerke berücksichtigt werden, die sich direkt auf die Komplexität der Benutzungsschnittstelle auswirken.

Zum einen bedarf es der Weiterentwicklung technischer Ein- und Ausgabegeräte, um das Ansprechen der menschlichen Sinne überhaupt gewährleisten zu können. Die nach wie vor im Mittelpunkt stehende visuelle Wahrnehmung wird durch stereoskopische Darstellungen mittels sogenannter Shutter-Brillen oder Head Mounted Displays weiter ausgebaut. Eine Positionsbestimmung des Betrachters mit Hilfe von Tracking-Systemen erlaubt zudem das Feststellen der Betrachterposition, sowie die Aufzeichnung von Hand- und Körperbewegungen.

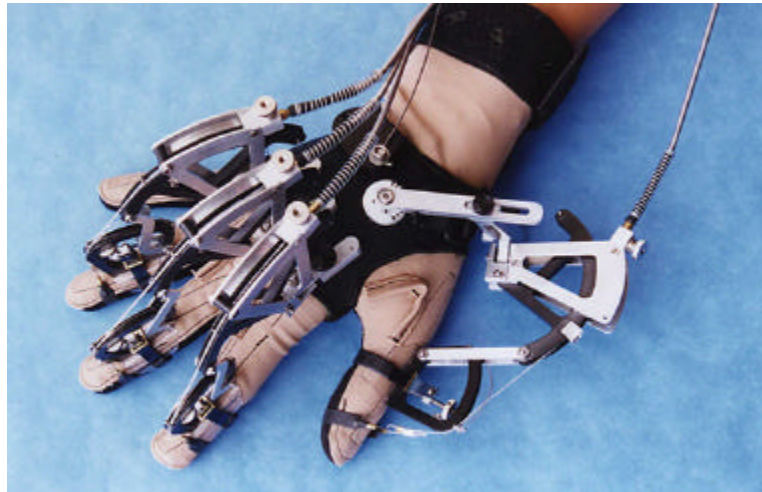


Abbildung 7: Datenhandschuh mit Force Feedback Mechanismen.

Das Erkennen von Gestik zur Realisierung konkreter Handlungsformen erfolgt mit Hilfe sogenannter Datenhandschuhe, die darüber hinaus auch die Möglichkeit der Krafterückkopplung (Force-Feedback) bieten (siehe Abbildung 7). Dabei wird durch das künstliche Erzeugen von Kräften der Eindruck von Berührung z.B. beim Greifen eines künstlichen Objektes simuliert.

Zusätzlich muß auf Seiten des Systemdesigns gewährleistet werden, daß ein in sich geschlossenes, einheitliches Gesamtmodell erzeugt wird ([Sno1994], S. 3). Die virtuelle Welt muß einen hohen Grad an Vollständigkeit erlangen, um gewohnte menschliche Verhaltensweisen und Umgangsformen berücksichtigen zu können.

Dazu bedarf es der umfassenden Modellierung grundlegender Gesetzmäßigkeiten, wie z.B. Gravitation, sowie der Repräsentation objektspezifischer Charakteristika und damit einhergehender Verhaltensweisen, wie z.B. der Repräsentation von Materie mittels Kollisionserkennung. Erst durch diese explizite Simulation von Regeln, Gesetzmäßigkeiten und Objektverhalten werden die Grundlagen sinnvollen Handelns geschaffen. Denn gerade die gegebenen Einschränkungen und kausalen Zusammenhänge einer Situation sind es, die eine schlüssige Interpretation der Verhältnisse ermöglichen und die Planung weiteren Vorgehens erlauben:

„The real world provides physical constraints that simplify interactions, but in an artificial reality, if two surgeon grab the brain and move it in different directions, what would happen?“ ([Fai1993], S. 16)

Die Bereitstellung einer solchen natürlichen Benutzungsschnittstelle im Rahmen einer eigenständigen virtuellen Welt impliziert deshalb, daß grundlegende menschliche Verhaltensweisen explizit modelliert und im System adäquat abgebildet werden müssen. Nur so kann der als entscheidend angesehene Schritt weg von einer passiven hin zu einer aktiven Form der Interaktion gewährleistet werden.

Die vollständige Integration von Benutzenden in eine virtuelle Modellwelt stellt die primäre Intention des Virtual Reality Konzeptes dar. Dabei kommt dem Ausbau der Benutzungsschnittstelle aufgrund der neuen, auf die menschlichen Sinne ausgerichtete Ein- und Ausgabemedien eine entscheidende Bedeutung zu. Die umfassende Bindung der menschlichen Wahrnehmung an das Interface soll gegenüber dem Ansatz der direkten Manipulation (siehe Kapitel 2.1) eine aktivere, erlebnisbasierte Umgangsform mit dem System ermöglichen, da die im Laufe der Zeit geprägten alltäglichen Handlungsmuster der Anwendenden in die Mensch-Maschine-Kommunikation einfließen können. Durch die auf diesem Wege zu erlangende Perspektive der Gegenwärtigkeit und aktiven Beteiligung am Geschehen verspricht man sich, eine natürliche und praxisorientierte Handhabung solcher Systeme realisieren zu können:

„However, if 3D interactive applications are to progress beyond simple walk-through visualizations or 3D versions of hyper-media interfaces, new 3D interaction metaphors and the appropriate software architectures with which to implement them must be discovered.“ ([Tur1998], S. 3)

Das Eintauchen in die virtuelle Welt verschärft aber gleichzeitig die bereits beim Direct Manipulation Interface aufgezeigte Notwendigkeit einer konsistenten, den Erwartungen der Benutzenden entsprechenden Realitätsabbildung. Der mit der Erweiterung der Benutzungsschnittstelle einhergehenden Komplexitätssteigerung muß auf Seiten der Modellwelt entsprochen werden. Die bereitgestellte künstliche Welt muß nicht nur aus einem objektiven, eher analytischen Standpunkt heraus schlüssig erscheinen, sondern bereits auf einer Ebene subjektiver, ggf. unterbewußter Eindrücke und Empfindungen den Anforderungen der Nutzenden gerecht werden.

So zeigt sich zum Beispiel, daß die fehlende taktile Rückkopplung bei der Navigation (Objekte können durchschritten werden) und dem Greifen von Gegenständen zu Orientierungslosigkeit und Verwirrung führt ([Pso1995], S. 4, [Bro1998], S. 14). Auch dem Anspruch einer realitätsnahen, überzeugenden Form der Visualisierung kann heutzutage nicht entsprochen werden, was sich teilweise durch das Auftreten der sogenannten Simulationskrankheit bemerkbar macht:

„Current microcomputers can realistically generate only a few thousand polygons per second, while it has been estimated that nearly a billion polygons per second may be needed for near realism.“ ([Pso1995], S. 3)

Zusätzlich zu den Problemen der Komplexitätsbewältigung und Abbildungsgenauigkeit, die in Zusammenhang mit der synthetischen Nachahmung von Realität stehen, ergeben sich generell schwerwiegende Konsequenzen für die Benutzenden in Bezug auf ihr eigenes Wahrnehmungsvermögen. Das Eintauchen in eine virtuelle Welt impliziert sowohl eine Trennung von dem realen, als auch sozialen und kulturellen Umfeld, sofern der Zugang zu einer virtuellen Welt ausschließlich über eine, die menschlichen Sinne ansprechende Benutzungsschnittstelle erfolgt. Gleichzeitig stellt dieses Interface das einzige Hilfsmittel dar, um sich in einer solchen computermedierten Welt orientieren und zurecht finden zu können, da ein Großteil der relevanten sinnlichen Reize vom Computer gesteuert werden.

Als Konsequenz folgt, daß sich die Benutzenden durch das Eintauchen in die virtuelle Welt gänzlich an das System ausliefern. Ihre Vorstellungen - also das zu entwickelnde mentale Modell – werden direkt durch das System kontrolliert, da die gesammelten Erfahrungen durch subjektive, aber künstlich erzeugte Eindrücke von der Schnittstelle vorgegeben werden können.

Der Rechner drängt sich in den Mittelpunkt des Geschehens. Er ersetzt die Wirklichkeit durch seine mathematisch kontrollierte, synthetisch erzeugte Illusion rationaler Welt:

„Thus, just as a microscope may amplify certain features of the world by reducing the vision field, VR may reduce our perceptions by becoming our perceptions. We become "absorbed" in our work and in our play.“ ([Laur1997], S. 6)

Gerade dieser Effekt der „Absorption“ in Kombination mit dem veränderten, virtualisierten Wahrnehmungsvermögen ist es, der einerseits die Faszination virtueller Welten ausmacht, andererseits aber auch die Gefahr der Entfremdung birgt. Sowohl das von einem VR-System generierte Umfeld, als auch die darin gesammelten Erkenntnisse sind lediglich imaginäre, gefilterte, synthetisch erzeugte Abstraktionen von Realität, denen immer auch ein Verlust an Informationen anhaftet. Diese Reduktion muß zwangsläufig zu einer unvollständigen, verfälschten Repräsentation von Welt führen. Die Abbildung eines Weltausschnittes impliziert die Definition eines festen Handlungsrahmens, der die möglichen Interaktionsspielräume in unüberwindbare Grenzen zwingt.

Demzufolge kann Virtual Reality nicht als adäquater Ersatz alltäglicher, „handfester“ Erfahrungen dienen. Vielmehr sollte dieses Konzept eingesetzt werden, um konkrete Anschauungen sinnlich nicht (er-) faßbarer Gegebenheiten zu vermitteln, d.h. den umgekehrten Weg einer Abstraktion verfolgen.

2.3 Real Reality Interfaces

Wie schon Virtual Reality zielt auch das Real Reality Konzept auf eine Vereinfachung des Umgangs mit Computern unter Einschluß der dritten Dimension ab. Die angestrebte Form der Interaktion beschränkt sich dabei nicht allein auf eine reine Kommunikation zwischen Mensch und Maschine, sondern nutzt explizit das direkte Umfeld der Benutzenden als Interaktionsraum. Es findet eine Verzahnung von Realität und Computersimulation statt, die eine Isolation der Partizipierenden von ihrer Umwelt verhindert und gleichzeitig die ausgeprägten taktilen Fähigkeiten des Menschen in den Vordergrund stellt.

Mit Hilfe geeigneter sensorischer Hilfsmittel wird das Hantieren der Anwendenden mit Objekten ihrer unmittelbaren Umgebung erfaßt und in ein virtuelles, der Realität entsprechendes Abbild übertragen. Aufgrund dieser Gleichschaltung von Realität und virtuellem Abbild ([Bra1997], S. 4) werden zwei unterschiedliche Sichtweisen ein und derselben Ausgangssituation geschaffen.

Somit steht es den Benutzenden frei, je nach Bedarf zwischen einer dieser Perspektiven zu wählen und sich so die Vorteile beider Ansichten gleicher Maßen zunutze zu machen:

„Über eine sensorisierte Hand werden dreidimensionale Form-, Lage- und Ortsänderungen realer gegenständlicher Modellelemente an eine rechnerinterne Modellwelt gekoppelt. Dadurch wird es möglich, mit Gegenständen der stofflichen Realität kommunikativ und operational umzugehen und den Rechner unterstützend zur Analyse und Variation im Rücken zu haben.“
([Bru1996b], S. 1)

Ziel ist es, die isolierte, eingrenzende Form der reinen Mensch-Maschine-Kommunikation aufzubrechen, um ein selbstbestimmtes, zielgerichtetes Agieren ohne prinzipielle technische Einschränkungen zu gewährleisten. Die ganzheitliche natürliche, sinnliche Wahrnehmung des Menschen in ihrer gesamten Komplexität und Vielschichtigkeit tritt gegenüber einer indirekten, vornehmlich visuell vermittelten Anschauung in den Vordergrund.

Dabei steht die konkrete gegenständliche Realität der Akteure im Zentrum des Geschehens. Der Computer als bisher dominantes, kontrollierendes Interaktionsmedium wird zugunsten wachsender Handlungsspielräume in den Hintergrund gedrängt. Ihm kommt zunächst eine passive, Geschehnis-protokollierende und -interpretierende Aufgabenstellung zu:

„Unser Ansatz verfolgt ein anderes Ziel: das des Erhalts von traditioneller Sinnlichkeit. Die Hand mit dem Datenhandschuh operiert mit realen Objekten, der Rechner steht als Hilfskraft im Rücken. Der Vorteil ist, daß alle originären, kreativen Akte des Menschen in der gewohnten anregenden Realität und kommunikativen Umgebung stattfinden, der Rechner aber vom Menschen im Hintergrund gehalten wird und bei Bedarf aktiviert werden kann.“
([Bru1993a], S. 18)

Die Fokussierung der Interaktion auf die Handlungsweisen der Benutzenden und dem daraus resultierendem Verhalten der gegenständlichen Objekte erfordert den Einsatz entsprechender technischer Hilfsmittel, die eine räumliche und dynamische Aufzeichnung erfolgter Tätigkeiten erlaubt. Das Erfassen der anfallenden Informationen erfolgt mittels geeigneter Eingabemedien wie z.B. Datenhandschuhen und Tracking-Systemen.

Anhand der übermittelten Informationen generiert der Rechner ein hinreichend realitätsnahes Abbild des aktuellen Szenarios, indem eine Interpretation erfolgter Gesten und Handlungsabfolgen unter Verwendung geeigneter Grifferkennungsalgorithmen und Analyseverfahren durchgeführt wird (siehe [Bra1994]). Auf diese Weise entsteht *gleichzeitig* ein physikalisch greifbares Produkt und ein dazugehöriges rechnerinternes, synthetisches Modell. Letzteres entspricht in seiner Form und Funktion weitgehend dem stofflichen Original. Erst bei diesem zweiten Schritt der Transformation vom Konkreten ins Abstrakte tritt – im Gegensatz zu virtuellen Umgebungen – eine Simplifizierung und Reduktion der ursprünglichen Informationsvielfalt ein.

Gerade diese in die Benutzungsschnittstelle einfließende zweigeteilte Repräsentation von Dingen stellt eine der elementarsten Grundlagen des Real Reality Ansatzes dar (siehe Abbildungen 8 und 9):

„In unserer Modellierungsumgebung gehen wir mit komplexen Objekten um, die wir mit unseren Händen manipulieren. Komplexe Objekte sind, in Anlehnung an den Begriff der komplexen Zahl, solche, die einen Realteil (gegenständlich und greifbar) und einen Imaginärteil (virtuell und im Rechner) haben.“
([Bru1998], S. 5)



Abbildung 8: Greifen eines realen Objektes.

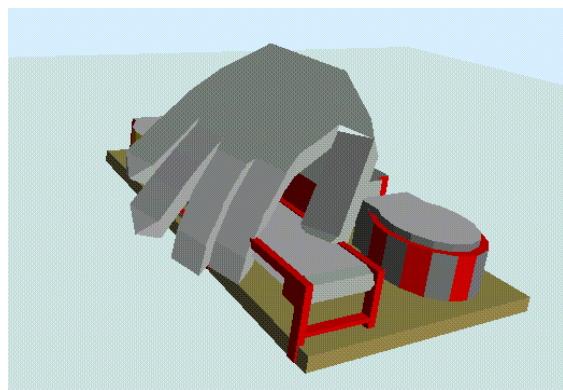


Abbildung 9: Synchrones Greifen des imaginären Gegenstücks im Virtuellen.

Im Vergleich zu den bisher dargelegten Konzepten der direkten Manipulation und virtuellen Welten findet hinsichtlich des Real Reality Ansatzes ein entscheidender Paradigmenwechsel statt. Die Einführung der doppelten Bedeutung von Objekten steht im Einklang mit der Aufspaltung der Benutzungsschnittstelle in einen aktiv ausführenden, sowie einen analytisch

betrachtenden Standpunkt. Dem entsprechend existieren zwei unterschiedliche, miteinander verflochtene Grundhaltungen, die wiederum entscheidende Veränderungen des Umgangs mit Computer nach sich ziehen.

Entgegen dem im Virtual Reality Konzept verfolgten Grundsatz der vollständigen Integration der Benutzenden wird hier eine eingeschränkte, indirekt vermittelte Wahrnehmung aufgrund der Nähe zur dinglichen Umgebung von vorn herein unterbunden. Die gesamten rezeptorischen Fähigkeiten des Menschen kommen dem angestrebten Prozeß der Erkenntnisgewinnung zugute, während eine für die computergestützten Simulation erforderliche Abstraktion von Welt erst im Zuge der Informationsaufnahme und -analyse am Rechner erfolgt. Da das daraus resultierende Ergebnis jedoch primär einer reflektierenden und analytischen Betrachtung dient, ist die so entstandene Kluft zwischen Realität und idealisierter Nachbildung beabsichtigt und unproblematisch.

Bezüglich des den Akteuren zur Verfügung stehenden Interaktionsraumes realisiert der Real Reality Ansatz ein Prinzip der Offenheit. Eine gekapselte, von der Wirklichkeit abgewandte künstliche Umgebung, wie sie in vielen virtuellen Welten angestrebt wird, ist hier nicht gefordert. Sie würde dem Akt des kreativen Denkens sogar im Wege stehen, zumal die Benutzungsschnittstelle solcher Systeme primär auf das Erkunden vorgegebener Zusammenhänge ausgerichtet ist und die derzeit verfügbaren Interaktionstechniken als eher primitiv zu bezeichnen sind ([Fol1987], S. 101). Statt dessen arbeitet die Schnittstelle quasi autark bzgl. des zu erzeugenden computerinternen Simulationsmodells:

„Der Mensch bleibt in seiner gewohnten gegenständlichen Welt situiert, erfährt die Sperrigkeit des Materials und macht weiterhin durch Berührung seine Erfahrung mit den Gegenständen. Der synchrone Übergang vom Gegenständlichen zum virtuellen Modell ermöglicht ihm, das Computermodell unmittelbar mit dem gegenständlichen Modell zu vergleichen und die unvermeidlichen Differenzen zwischen beiden deutlich zu erkennen.“ ([Rob1997], S. 171)

Auch hinsichtlich der Bildung und Verifikation eines mentalen Modells bietet Real Reality gegenüber Virtual Reality und Direct Manipulation Konzepten eine vorteilhafte Ausgangslage. Die durch den direkten Umgang mit dem Material gewonnenen subjektiven Eindrücke eröffnen den unmittelbaren Zugang zu einem konkreten Sachverhalt und tragen gleichzeitig zum Gewinn einer durchgehend wirklichkeitsgetreuen Anschauung bei:

„Der sinnlichen Wahrnehmung und Überprüfung realstofflicher Phänomene wird eine zentrale Bedeutung für das Begreifen und Beurteilen komplexer Zusammenhänge zuerkannt.“ ([EUG@], Motivation)

Das manuelle Greifen steht demzufolge in unmittelbarem Zusammenhang mit dem Vorgang des gedanklichen „Begreifens“. Die Erfahrung von Sinnlichkeit und die unscheinbare Existenz alltäglicher Gesetzmäßigkeiten ist dem menschlichen Denken und Handeln vorangestellt und prägt die Gestalt des individuellen, mentalen Weltbildes erheblich.

Die dem Real Reality Ansatz zugrunde gelegte, am konkreten Objekt orientierte Interaktion macht sich gerade diesen Umstand zunutze, um bei den Benutzenden eine adäquate mentale Modellvorstellung hervorzurufen. Das real existierende Modell stellt quasi eine „Verständnis-schnittstelle“ zwischen der individuellen Vorstellung der Modellierenden und der virtualisierten, rationalen Modellwelt des Computers dar.

Im Gegensatz dazu kann keiner der vorangegangenen Ansätze eine solche Analogie zwischen konkreter Handlung und daraus gewonnenem Verständnis realisieren, da in beiden Fällen der Computer als Kommunikationsmedium, also vermittelnde Instanz, eingesetzt wird und dieser Umstand immer auch eine Entfremdung von Welt mit sich bringt.

Als letzter Aspekt des Real Reality Konzeptes soll an dieser Stelle die dem Ansatz inhärente Unterstützung von Gruppenarbeit dargestellt werden:

Der Umstand, daß sich Anwendende nicht zum Zwecke der Kommunikation mit dem Rechner gegenüber ihrem sozialen Umfeld abschotten müssen, macht sich gegenüber Virtual Reality auch an dieser Stelle vorteilhaft bemerkbar. Die Möglichkeit des Gestikulierens und Hantierens mit physikalischen Gegenständen führt sogar dazu, daß selbst der in Zusammenhang mit reiner Bildschirmarbeit auftretende Isolationseffekt vermieden werden kann.

Die Ausrichtung der Interaktion auf das unmittelbare Umfeld der Anwendenden birgt vielmehr das Potential einer ausgeprägten Kooperation aller Beteiligten. Das gemeinsame „Schaffen“ substanzieller Konstruktionen erlaubt parallel dazu einen angeregten, problemorientierten Austausch von Wissen und Erfahrungen der Beteiligten untereinander. Dabei besteht jederzeit die Möglichkeit der direkten Bezugnahme auf das vorliegende Szenario. Auf eine einheitliche, tiefgreifende Wissensbasis der Teilnehmer kann deshalb prinzipiell verzichtet werden. Vielmehr wird auch jenen eine anschauliche Diskussionsgrundlage geboten, die noch keine ausgedehnten Kenntnisse in dem gegebenen Themengebiet vorweisen können. Ihnen wird sogar die Chance eröffnet, sich im Diskurs mit anderen ein Bild der vorgefunden Zusammenhänge zu erarbeiten:

„Der gesamte Vorgang der Modellerstellung kann gemeinschaftlich mit mehreren Personen, die sich gegenseitig ergänzen, Ideen einbringen oder Kritik äußern können, erfolgen. Damit bietet Real Reality nicht nur eine alternative Eingabeschnittstelle zum Computer, sondern darüber hinaus ein Kommunikationsmedium für mehrere Benutzer, das mit einem Rechner gekoppelt ist. ... Damit stehen für ein und dieselbe Aufgabe völlig verschiedene Perspektiven auf eine Problematik zur Verfügung. Je nach persönlicher Präferenz und abhängig von der jeweiligen Zielsetzung, kann eine dieser Perspektiven zum Verständnis einer Problemsituation herangezogen werden.“ ([Bra1997], S. 7)

Eine völlig andere Ausgangslage besteht derzeit im Bereich Virtual Reality. Die bereits angesprochenen unterentwickelten Interaktionstechniken in Verbindung mit der funktionalen und strukturellen Komplexität der simulierten Welten lassen eine echte Kooperation zwischen den Partizipanden nicht zu:

„Communication in virtual worlds is still quite primitive. Users can communicate through objects they manipulate and by some gestures, although it is hard to separate communicative gestures from the command and control gestures in most VR systems.

Direct Communication between users is even harder. ... In today's VR worlds, we cannot even leave each other written notes. This significantly limits the utility of most virtual worlds.“ ([Wex1993], S. 39)

Zusammenfassend kann festgestellt werden, daß sich eine Vielzahl von Möglichkeiten aus dem Konzept des computergestützten sinnlichen Umgangs mit Dingen der stofflichen Welt ableiten lassen.

Voraussetzung allerdings ist, daß man die grundlegende Basis der physischen und physikalischen Wirklichkeit nicht verläßt, um in abstrakte, menschenfeindliche oder rein theoretische Gebiete vorzudringen. Hierfür scheint mir Virtual Reality der geeignetere Ansatz zu sein.

Die genauere Betrachtung der Benutzungsschnittstelle jedoch zeigt, daß die Idee, den „Computer im Rücken“ zu haben, den Anforderungen einer direkten, an den menschlichen Bedürfnissen ausgerichteten Interaktion weitestgehend gerecht werden kann. Der Virtual Reality Ansatz hingegen scheitert - zumindest aus heutiger Sicht - an dem Versuch, den Computer durch die Kontrolle der menschlichen Sinne weiter in den Mittelpunkt des Geschehens zu drängen.

2.4 Alternativansätze und Projekte

Nachdem in den vorangegangenen Teilen dieses Kapitels die in Zusammenhang mit dieser Arbeit wesentlichen Konzepte diskutiert wurden, möchte ich im weiteren Verlauf dieses Abschnitts einige weitere interessante Ansätze und Projekte exemplarisch vorstellen. Alle diese Entwicklungen stehen in einer mehr oder minder engen Beziehung zu dem hier entwickelten Interaktionskonzept - sei es aufgrund der Bestrebung, eine intuitive Umgangsform mit dem Computer unter Zuhilfenahme realer Gegenstände zu realisieren, oder an Baukästen angelehnte Systeme zu entwickeln, die komplexe Strukturen anschaulich repräsentieren.¹

¹ Auf eine detaillierte Erörterung soll an dieser Stelle jedoch verzichtet werden, da dies den Rahmen der Arbeit sprengen würde. Insoweit wird auf weiterführende Literatur verwiesen.

2.4.1 Greifbare Benutzungsschnittstellen

Die in Zusammenhang mit den heute üblichen Graphical User Interfaces stehende, am Computer ausgerichtete Form der Interaktion mit Bildschirm, Tastatur und Maus bietet – wie bereits dargelegt – nicht die benötigten Freiräume für eine intuitive Form der Interaktion. Greifbare Benutzungsschnittstellen versuchen diesen Einschränkungen entgegen zu wirken, indem sie neue, spezialisierte Eingabegeräte und –techniken entwickeln. Durch eine Verlagerung der Datenmanipulation ins Gegenständliche erhofft man sich, die ausgeprägten Fähigkeiten des menschlichen Tastsinns und der räumlichen Wahrnehmung besser ansprechen zu können. Die hoch entwickelte, durch das tägliche Leben geprägte Motorik der Benutzenden soll in die Schnittstelle integriert werden, um eine natürliche, intuitive Form der Interaktion zu gewährleisten.

Im Unterschied zum Real Reality Konzept setzen solche greifbaren Benutzungsschnittstellen ein Interaktionsmodell voraus, welches die jeweilige Bedeutung der eingesetzten, in ihrer Funktionalität spezialisierten Gegenstände festlegt. Die Resultate einer Manipulation sind von der Interpretation des Systems abhängig. Sie werden nicht durch die physikalischen Objekteigenschaften selbst bestimmt, d.h. die Interaktion mit diesen Artefakten steht nicht im Einklang zur Realen Welt. Ihr haftet nach wie vor ein symbolischer Charakter an, der eine Differenz zwischen technischer und realer Auswirkung einer Aktion aufrecht erhält.

Die Mehrzahl der hier vorgestellten Ansätze läßt sich dem Anwendungsfeld „Augmented Reality“¹ zuordnen. Dahinter verbirgt sich das generelle Anliegen, die Reale Welt um virtuelle Perspektiven und Gesichtspunkte zu erweitern. Dabei soll - im Gegensatz zu Virtual Reality - vermieden werden, daß die Benutzenden ihr gewohntes Umfeld aus den Augen verlieren. Vielmehr wird angestrebt, den Computer auf eine zurückhaltende Art und Weise in die Welt der Benutzenden zu integrieren. Der Mensch verweilt in seinem realen Umfeld, kann seine sozialen Beziehungen aufrecht erhalten und nutzt nicht zuletzt seine im Laufe der Zeit erworbenen und als selbstverständlich erachteten, natürlichen Handlungsfähigkeiten.

Diese Voraussetzungen zugrundelegend wird angestrebt, das Potential des alltäglichen Handelns mit in die Mensch-Maschine-Kommunikation einfließen zu lassen. Reale Objekte werden um elektronische Charakteristika oder Funktionalitäten erweitert, ohne dabei ihre physikalischen Eigenschaften einbüßen zu müssen. Erreicht wird diese Ausweitung mit Hilfe unterschiedlicher Projektionstechniken, wie z.B. durch den Einsatz einfacher Leinwandprojektoren, sogenannter Shutter-Brillen oder „See-Through-Devices“² (siehe [Wel1993a]).

¹ Augmented: vergrößern, erweitern, steigern

² Mit Hilfe von Shutter-Brillen läßt sich ein dreidimensionaler, räumlicher Eindruck erzeugen, indem in hoher Frequenz jeweils eines der beiden Brillengläser undurchsichtig geschaltet wird. Synchron dazu liefert ein Computer Bilder, die jeweils für den Blickpunkt des linken oder rechten Auges berechnet wurden.

Bei See-Trough-Devices handelt es sich ebenfalls um spezielle Brillen. Sie bieten die Möglichkeit, auf Teile der

2.4.1.1 Graspable User Interface

Ausgehend von einer grundsätzlichen Unterscheidung zeitlich und örtlich gebundener Eingabegeräte stellt George Fitzmaurice ([Fit@])¹ das Konzept des „*Graspable User Interface*“ vor. Die von ihm vorgenommene Differenzierung der Manipulationshilfsmittel bezieht sich dabei auf das Verhältnis des Eingabegerätes zu der ihm zugeschriebenen, systeminternen Funktionalität.

„*Time-multiplexed*“ bezeichnet in diesem Zusammenhang solche Hilfsmittel, deren Funktionalität sich im Zuge zeitlicher Abläufe verändert. Die vom System ausgeführte Funktion ist abhängig von dem konkreten Kontext, in dem das Gerät als Werkzeug eingesetzt wird. Daraus folgt, daß diese breitbandig ausgelegten Eingabemedien eine Vielzahl von Aufgaben übernehmen können, sich aber gerade deshalb nicht sehr effizient handhaben lassen. Die Maus ist ein Beispiel eines solchen Eingabegerätes.

Das Gegenteil dazu stellen sogenannte „*space-multiplexed*“ Gerätschaften dar. Solche bekommen bezüglich des zeitlichen Verlaufs eine eindeutige, unveränderliche Funktion zugewiesen. Zusätzlich zu dieser funktionalen Konsistenz belegt jedes dieser Geräte zu einem bestimmten Zeitpunkt eine räumliche Position, d.h. es gibt einen eindeutigen Ort, an dem sich das Instrument befindet. Das Lenkrad eines Autos wäre ein Beispiel für ein solches Steuerungsgerät.

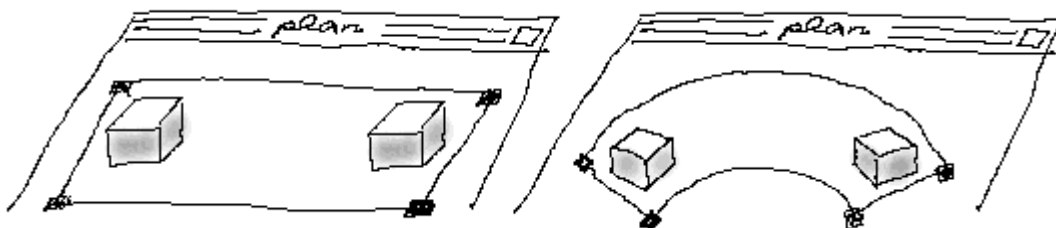


Abbildung 10: Verbiegen eines grafischen Quaders mit Hilfe zwei verankerter Klötze

Aus dieser Unterscheidung heraus entwickelt Fitzmaurice das „Graspable User Interface“. Im Rahmen der prototypisch realisierten Anwendung „Active Desk“ (siehe Abbildung 10) werden physikalische Klötze als Interaktionshilfsmittel zur Manipulation zweidimensionale Geometrien eingesetzt. An diesen Klötzen werden Eigenschaften virtueller Objekte verankert und somit örtlich gebunden. Solche instanziierten Funktionen vereinfachen den Umgang mit einem System aufgrund folgender Zusammenhänge:

ansonsten durchsichtigen Gläser computergenerierte Bilder zu projizieren, die dann an entsprechender Stelle das Realbild überdecken.

¹ Zugrunde gelegte und weiterführende Literatur: [Fit1995], [Fit1996], [Fit1997].

- Die feste Bindung einer Funktion an ein physikalisches Objekt erlaubt eine Vielzahl spezialisierter, auf den Anwendungskontext bezogene Eingabemedien.
- Die physikalischen Funktionsrepräsentanten fördern die ausgeprägten motorischen Fähigkeiten der Benutzenden.
- Mehrere Objekte können gleichzeitig mit beiden Händen manipuliert werden.
- Mehrere Personen können gleichzeitig Manipulationen durchführen.
- Aufgrund der funktionalen Persistenz der Klötze kann deren räumliche Anordnung von den Anwendern als kognitive Hilfestellung genutzt werden. Antizipierte Handlungen werden schneller auf die aktuelle Anwendungssituation übertragbar.

2.4.1.2 Natural User Interface

Mit dem „Natural User Interface (NUI)“ verfolgt M. Rauterberg ([Rau@])¹ ebenfalls eine Art des Umgangs mit Computern, die sich von der klassischen Form des Graphical User Interfaces abhebt. Wie auch im Falle des „Active Desk“ ist hier die Benutzungsschnittstelle weitgehend vom Computer getrennt, so daß die klassische Form der computerzentrierten Interaktion entfällt. Statt dessen wird angestrebt, die Mensch-Maschine-Schnittstelle so zu gestalten, daß ein natürliches Verhalten der Benutzenden durch das Einbeziehen ihrer Körper (-teile) ermöglicht wird.

Umgesetzt wurde dieses Konzept in Form eines Prototypen, dem „Digital Desk“. Wie auch in vergleichbaren Ansätzen (Siehe [Wel1993b]) dient hier ein horizontaler Arbeitsbereich als Interaktionsebene. Mit Hilfe einer digitalen Kamera werden Handbewegungen sowie die Lage eines Klötzchens – dem „universal interaction handler“ (siehe Abbildung. 11) - auf der Arbeitsfläche erfaßt und als Eingabeinformationen vom Computer verarbeitet. Eine direkte Informationsrückkopplung erfolgt mit Hilfe einer Projektion der Computerausgabe auf den Arbeitstisch. Anhand dieser Projektion können Veränderungen der Szene direkt während der Interaktion verfolgt werden.

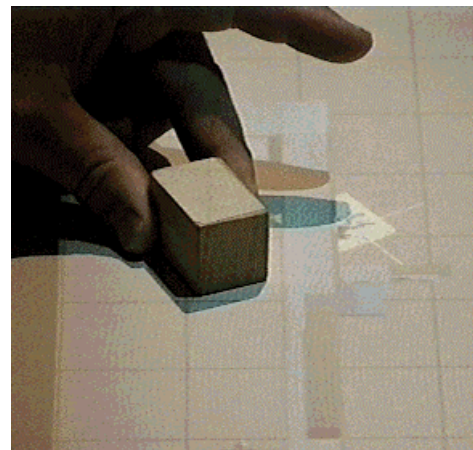


Abbildung 11: Ein Holzbaustein dient hier als Interaktionshilfsmittel, um z.B eine virtuelle Kamera zu steuern.

¹ Zugrunde gelegte und weiterführende Literatur: [Rau1997a], [Rau1997b], [Rau1998]

Zusätzlich zu dieser „working area“ existiert eine sogenannte „communication area“. Hierbei handelt es sich um eine vertikal angebrachte Leinwand, auf die zusätzliche Informationen projiziert werden können. Zum Beispiel kann im Kontext einer Layoutplanung zur Innenraumgestaltung eine Grundrißprojektion auf die Arbeitsfläche erfolgen, wo hingegen der Kommunikationsbereich für eine dreidimensionale Darstellung des aktuellen Arrangements genutzt wird (siehe Abbildung 12).



Abbildung 12: Aktionsraum des „Digital Desk“-Systems.

Folgende entscheidenden Fähigkeiten des „Digital Desk’s“ werden von M. Rauterbach angeführt:

1. Die Projektion virtueller Objekte in Form von Bildern auf die Arbeitsfläche oder darauf befindlicher Objekte.
2. Das System reagiert auf die Veränderung realer Objekte, wie zum Beispiel Stifte oder Hände.
3. Es erlaubt die Interpretation auf einer angemessenen sematischen Ebene.

2.4.2 Sensored Objects

Ein weiterer Ansatz der Integration des Computers in die dingliche Welt besteht in der Idee, Gegenstände mit einem räumlichen „Bewußtsein“ zu versehen. Durch die geeignete Sensorisierung von Objekten wird z.B. die Möglichkeit geschaffen, räumliche Zusammenhänge zu erkennen oder unterschiedliche Zustände der direkten Umgebung aufzunehmen. Eine Analyse der Menge von Einzelinformationen liefert dann in einem zweiten Schritt ein virtuelles Bild der Umgebung.

Entgegen dem Real Reality Konzept, welches sich die Hand der Akteure als universelle Informationsquelle zunutze macht, erfolgt in diesem Falle ein „Vermessen von Welt“. Nicht das aktive Auslösen von Veränderungen durch den Menschen steht hier im Mittelpunkt des Interesses, sondern das daraus resultierende „objektive“ Ergebnis.

2.4.2.1 AlgoBlock

Bei AlgoBlock¹ handelt es sich um eine an Logo angelehnte Programmiersprache mit einer gegenständlichen Ausprägung in Form von Blöcken. Jedem verfügbaren Befehl der Sprache entspricht ein greifbarer Quader, der über eine Schnittstelle mit anderen verbunden werden kann. Durch das Zusammenfügen einer Anzahl solcher Klötze entsteht eine Sequenz von Kommandos. Ein auf diese Weise erstelltes Programm erlaubt die Steuerung eines, auf einem Computerbildschirm dargestellten U-Bootes.

„AlgoBlock“ wurde entwickelt, um Schülern die Entwicklung von Lösungsverfahren am Beispiel der Programmierung nahe zu bringen. Dabei liegt ein Hauptaugenmerk auf der Unterstützung praxisorientierter Gruppenarbeit. Das System ist als ein „open tool“ entworfen worden, welches es erlaubt, die dynamischen Prozesse eines kooperativen Vorgehens offen darzulegen. Die Möglichkeiten, Handlungen der Partizipierenden zu studieren, ihre Gedankengänge anhand ihres Verhaltens nachvollziehen zu können und den aktuellen Stand der Entwicklung jederzeit vor Augen zu haben, sind entscheidende Kriterien dieses Konzeptes. Dabei kommt den, der Gruppe zur Verfügung stehenden Programmierbausteinen die Funktion eines realitätsbezogenen Interaktionsmediums zu, das die aktive Kooperation der Gruppenmitglieder untereinander überhaupt erst ermöglicht und darüber hinaus eine aktivitätssteigernde Wirkung ausüben soll (siehe Abbildung 13).



Abbildung 13: Kinder programmieren mit AlgoBlock.

2.4.2.2 LEGOsheets

LEGOsheets² steht in engem Zusammenhang mit dem am MIT³ entwickelten „LEGO Programmable Brick“ und bildet in Kombination mit diesem eine Programmier- und Experimentierumgebung. Diese soll Kindern ab einem Alter von acht Jahren eine stufenweise, praxisorientierte Einführung in die Programmierung einfacher, regelbasierter Systeme bieten.

¹ Zugrunde gelegte und weiterführende Literatur: [Suz1995], [Bra1994]

² Zugrunde gelegte und weiterführende Literatur: [Gin1995], [Res1994]

³ Massachusetts Institute of Technology, Boston, USA



Abbildung 14: LEGOsheets zur Simulation und Programmierung des Brick.

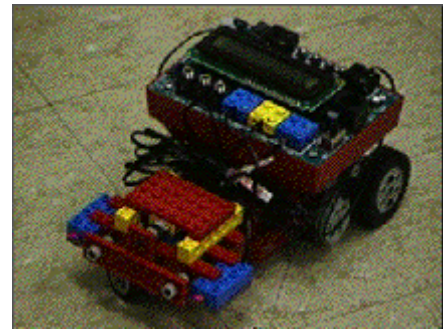


Abbildung 16: Lego Pet

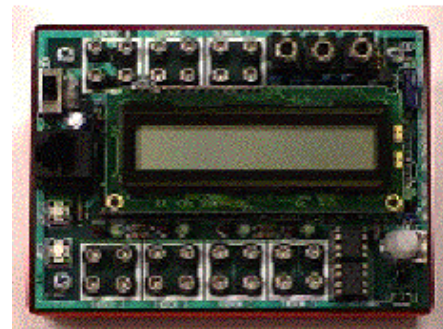


Abbildung 15: Programmable Brick

Bei dem „LEGO Programmable Brick“ handelt es sich um einen kleinen eigenständigen Computerbaustein, der es in Kombination mit diversen Sensoren und Effektoren erlaubt, Eigenschaften und Verhalten technischer Gebilde zu bestimmen und zu steuern (siehe Abbildung 15). Der Baustein ist so konstruiert, daß er sich mit herkömmlichen LEGO Elementen verbinden läßt und so in eine Vielzahl unterschiedlicher Modellkonstruktionen integriert werden kann (siehe Abbildung 16).

LEGOsheets hingegen enthält neben einer visuellen Programmiersprache einen grafisch dargestellten Simulationsbereich, der einen virtuellen „Programmable Brick“ und die dazugehörigen, durch Icons repräsentierten Sensoren und Effektoren umfaßt (siehe Abbildung 14). Im Rahmen dieser Umgebung besteht nun die Möglichkeit, eine virtuelle, technische Konstruktion zu entwickeln und deren Verhalten durch den Simulator spielerisch zu ergründen.

Zusätzlich zu dieser rein theoretischen Betrachtung kann ein realer „Programmable Brick“ an das System angeschlossen werden, um die Funktionalität des Gerätes direkt in der realen Umgebung überprüfen zu können. Die Möglichkeit, durch Umschalten nach Belieben zwischen virtuellen und realen Sensoren und Effektoren wechseln zu können, realisiert auf diese Weise den gewünschten experimentellen Übergang zwischen Theorie und Praxis.

Das hinter dieser Entwicklung stehende Konzept zielt im Wesentlichen darauf ab, Kindern einen graduellen Übergang zwischen einer manuell, anschaulichen und der abstrakt theoretischen Perspektive bereitzustellen. Dabei ist interessanter Weise festzustellen, daß das von den Autoren dargestellte Vorgehen der Kinder von dem theoretischen Teil ausgehend hin zum Gegenständlichen erfolgt und nicht umgekehrt.

Kapitel 3

Synchrones Modellieren

Nachdem in den vorangegangenen Kapiteln die theoretischen Grundlagen dieser Arbeit abgesteckt wurden, konzentrieren sich die nun folgenden Kapitel auf die Darstellung praxisrelevanter Aspekte. Bevor ich aber konkret auf die von mir entwickelten Algorithmen und technischen Fragestellungen der Implementierung eingehe, erfolgt zunächst eine Erläuterung grundlegender Vorstellungen, Rahmenbedingungen und Zielsetzungen.

Nach einer Einführung der Grundidee wird in diesem Kapitel eine Analyse der Anwendungsstruktur vorgenommen, um die generell wünschenswerten Anforderungen an das Programm festlegen zu können. Darüber hinaus werden auch praxisrelevante Kriterien mit in Betracht gezogen, um bei der theoretischen Analyse die Frage der Machbarkeit nicht gänzlich aus den Augen zu verlieren.

3.1 Die Grundidee

Zu der Zeit, als ich mich nach einem geeigneten Anwendungsgebiet für meine Diplomarbeit umsah, wurden am Forschungszentrum Arbeit und Technik der Universität Bremen bereits Entwicklungen hinsichtlich des Real Reality Konzeptes vorgenommen. Im Rahmen des Projektes RUGAMS¹ befand sich ein System in Arbeit, welches die Layoutgestaltung einer teils realen, teils simulierten Produktionsanlage mit Hilfe physikalischer Modelle erlauben sollte.

Dabei kamen u.a. verschiedene Arten von Förderband-Modellen zum Einsatz, die für den Anwendungszweck jedoch zunächst in zeitraubender Kleinarbeit aus Fischertechnik-Bausätzen anhand der vorliegenden Bauanleitung zusammengesetzt werden mußten. Nachdem diese Aufgabe abgeschlossen war, konnte man mit den Modellen nun endlich die geplanten Produktionsstraßen aufbauen und in Betrieb nehmen. Eine adäquate Darstellung der beeindruckenden Konstruktionen am Bildschirm blieb zur Ernüchterung aller Beteiligten jedoch zunächst aus.

¹ Rechnergestützte Übergänge zwischen gegenständlichen und abstrakten Modellen produktionstechnischer Systeme.

Also ließ man die „Spielzeuge“ – wie die Modelle von unwissenden Dritten oftmals titulierte wurden – wieder auf dem Arbeitstisch zurück und wandte sich dem Computer zu. Erneut stellte man sich der Aufgabe des Nachbildens, diesmal allerdings im virtuellen Raum und unter Zuhilfenahme geeigneter Software. Erst mit Fertigstellung der zweiten, geometrischen Repräsentation konnte man sich schließlich dem eigentlichen Gegenstand der Layoutgestaltung zuwenden.

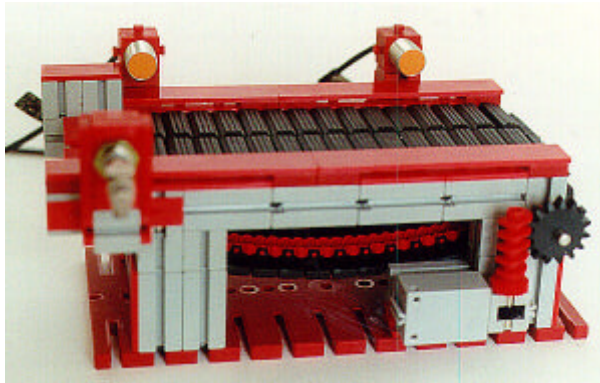


Abbildung 18: Modell eines Förderbandes aus Fischertechnik-Bausteinen

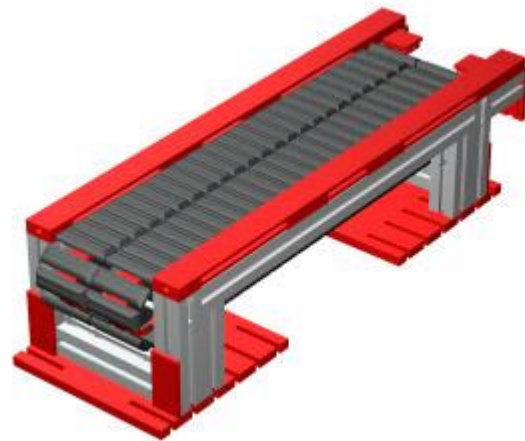


Abbildung 17: Grafische Repräsentation eines Förderbandes.

Schon fast zwangsläufig entsprang aus dieser Situation die Idee, man könne doch bereits den Montageprozeß des gegenständlichen Modells dazu nutzen, gleichzeitig ein passendes, rechnerinternes Gegenstück zu generieren (siehe Abbildungen 17 und 18). Vorteilhaft wäre nicht nur, daß man den Zeitaufwand der zweiten Konstruktionsprozessen eingespart hätte, man würde darüber hinaus ein in seinem Grundaufbau dem Original ähnelndes Modell erhalten. Zusätzlich könnte der Verlauf des Modellaufbaus im Computer archiviert und zu einem späteren Zeitpunkt als Rekonstruktionshilfsmittel wieder abgerufen werden. Damit war die Grundidee dieser Diplomarbeit entstanden.

3.2 Generelle Anforderungen

Nachdem die grobe Zielrichtung der hier vorgestellten Arbeit festgelegt war, galt es in weiteren Schritten eine konkrete Definition des Anwendungsspektrums vorzunehmen. Um ein hinreichend breites Einsatzfeld berücksichtigen zu können, sollte eine Applikation, die sich dem Gebiet des Gestaltens von Modellen im Realen und Virtuellen widmet, bestimmten Kriterien genügen. Generell sollte ein Grad an Flexibilität realisiert werden, der einen universellen Einsatz des Systems ermöglicht und darüber hinaus anwendungsspezifischen Ansprüchen gerecht werden kann.

Für einen Prototypen, der speziell auf die Verwendung von Baukastensystemen ausgerichtet ist, ergeben sich in dieser Hinsicht einige allgemeine Anforderungen, die es zu berücksichtigen gilt:

- Unterschiedliche Baukastensysteme lassen sich aufgrund ihrer vielfältigen Gestalt in unterschiedlichsten Anwendungsgebieten einsetzen. In Abhängigkeit von der Bauweise und dem angestrebten Darstellungszweck eines Modells gilt es zu entscheiden, welches System für welchen Bereich zweckmäßig oder eher ungeeignet erscheint.
So werden für eine architektonische Konstruktion beispielsweise vornehmlich solche Bausteine Verwendung finden, die die Modellierung flächiger, geschlossener Strukturen erlaubt. Demgegenüber wird bei mechanischen oder elektronischen Konstruktionen eher auf die Darstellungsmöglichkeit funktionaler Aspekte Wert gelegt. Deshalb kommen in diesem Falle bevorzugt solche Bausteinsätze zum Einsatz, die einen Schwerpunkt auf die Repräsentation technisch-operationaler Zusammenhänge setzen.
In einer Applikation, die den Einsatz unterschiedlicher Konstruktionsmittel unterstützen und nicht behindern soll, muß daher ein möglichst breites Spektrum verschiedenartiger Baukastensysteme Berücksichtigung finden.
- Der Funktionsumfang eines Baukastensystems darf nicht unbeabsichtigt aufgrund seines virtuellen Pendants eingeschränkt werden.
Die Anwendung muß so ausgelegt sein, daß sie die Mächtigkeit eines Modellbaukastens auf das rechner-interne Gegenstück umfassend abzubilden vermag. Die den Bausteinen anhaftende Methodik muß im Rahmen der Simulation vollständig reproduzierbar sein, um Einschränkungen in der prinzipiellen Kombinatorik der Elemente bereits auf struktureller Ebene ausschließen zu können.
- Der Einsatz von Modellbaukästen ist primär durch den Vorgang des Gestaltens am Gegenstand geprägt. Dieser Prozeß sollte generell nicht beeinträchtigt werden, d.h. eine Unterbrechung des eigentlichen Arbeitsablaufes durch die Anwendung gilt es zu vermeiden. Anwendende sollen sich gänzlich auf ihre kreativen Tätigkeiten konzentrieren können, ohne sich zwischenzeitlich bewußt dem System zuwenden zu müssen.
- Der konkrete Arbeitsfluß der Anwendenden sollte nicht gestört werden. Der Bedarf an Verarbeitungszeit während des Modellierens - verursacht durch die notwendige Aufbereitung der anfallenden Informationen - sollte soweit reduziert werden, daß keine signifikanten zeitlichen Verzögerungen auftreten.
Die Analyse und Interpretation simulationsrelevanter Daten muß unauffällig im Hintergrund, d.h. in Echtzeit erfolgen.

Um den hier dargestellten Anforderungen im Rahmen eines Programmentwurfs und der daran anknüpfenden Systementwicklung gerecht werden zu können, bedarf es zunächst einer näheren Betrachtung des eigentlichen Anwendungsgebietes.

3.3 Analyse und Eingrenzung

Die umfassende Analyse des Modellierungsvorgangs und der daran beteiligten Objekte soll aufschluß darüber geben, welche strukturellen Komponenten und gegenseitigen Abhängigkeiten existieren, auf dessen Grundlage der Prozeß des Konstruierens mit Hilfe des Computers nachvollzogen werden kann. Es muß entschieden werden, welchen Informationen eine maßgebliche Bedeutung hinsichtlich der Programmentwicklung zukommt und in welcher Form diese zu interpretieren sind.

Bereits auf den ersten Blick lassen sich zwei strukturell unterschiedliche Bereiche des vorgegebenen Themenkomplexes ausmachen:

- Bei dem Vorgang des Modellaufbaus handelt es sich um eine dynamische, variationsreiche Prozedur, die es auf geeignete Weise zu verfolgen gilt.
- An diesem Vorgang sind Objekte beteiligt, die in ihrer Gesamtheit das Regelwerk einer festgelegten und komplexen Montagesystematik definieren und somit dem Baukastensystem seinen speziellen Charakter verleihen.

Ausgehend von dieser generellen Unterscheidung sollen nun einige weitere Betrachtungen vorgenommen werden, um einen besseren Einblick in die Beschaffenheit dieser beiden Bereiche zu erhalten.

3.3.1 Objektstrukturen

Die an dieser Stelle entwickelte Klassifizierung bezieht sich zunächst auf die an dem Konstruktionsprozeß beteiligten materiellen Komponenten. Sie soll den gegenständlichen Aspekt des Geschehens erfassen und damit Aufschluß über dessen Bedeutung im Gesamtkontext der Modellierung geben.

3.3.1.1 Bausteintypen

Baukästen bestehen im allgemeinen aus einer Reihe von Elementen, die sich auf unterschiedliche Weise miteinander verbinden lassen. Aufgrund dieser kombinatorischen Eigenschaften besteht die Möglichkeit, daraus unterschiedlich komplexe Gebilde, also Strukturen höherer Ordnung, zusammensetzen zu können. Der Gestalt der einzelnen Bausteine kommt in diesem Zusammenhang eine zentrale Bedeutung zu, denn diese bestimmt, welche Verbindungen zwischen Objekten unter welchen Bedingungen realisierbar sind.

Das Erscheinungsbild der Bausteine spielt dabei eine entscheidende Rolle. Die Elemente eines einfachen Satzes Bauklötze können zum Beispiel nur dann aufeinander gestellt werden, wenn sich deren angrenzende Kanten in einer horizontalen Ebene befinden. Anderenfalls würde der obere Stein keinen Halt finden und wieder zu Boden fallen.

Auch müssen die Bausteine in ihren Proportionen zueinander passen. Zum Beispiel erzeugen zwei aneinander gestellte Quader mit unterschiedlicher Kantenlänge einen Absatz. Dieser verhindert, daß ein dritter Stein pyramidenartig auf die Mittelkante der Blöcke gesetzt werden kann, ohne daß er sich zwangsläufig zur flacheren Seite der Kante neigt. Eine einheitliche Dimensionierung der einzelnen Bausteine ist deshalb eine der elementaren Eigenschaften eines jeden Baukastensystems.

Während sich ein Satz Bausteine in der Regel aus einer relativ geringen Anzahl geometrischer Grundformen zusammensetzt, übersteigt die Masse der Konstruktionselemente technisch orientierter Systeme schnell die Zahl des Überschaubaren. Aber trotz dieser Menge unterschiedlichster Formen haben alle diese Elemente eine grundlegende Eigenschaft gemeinsam: Ein einheitliches Kombinationsprinzip.

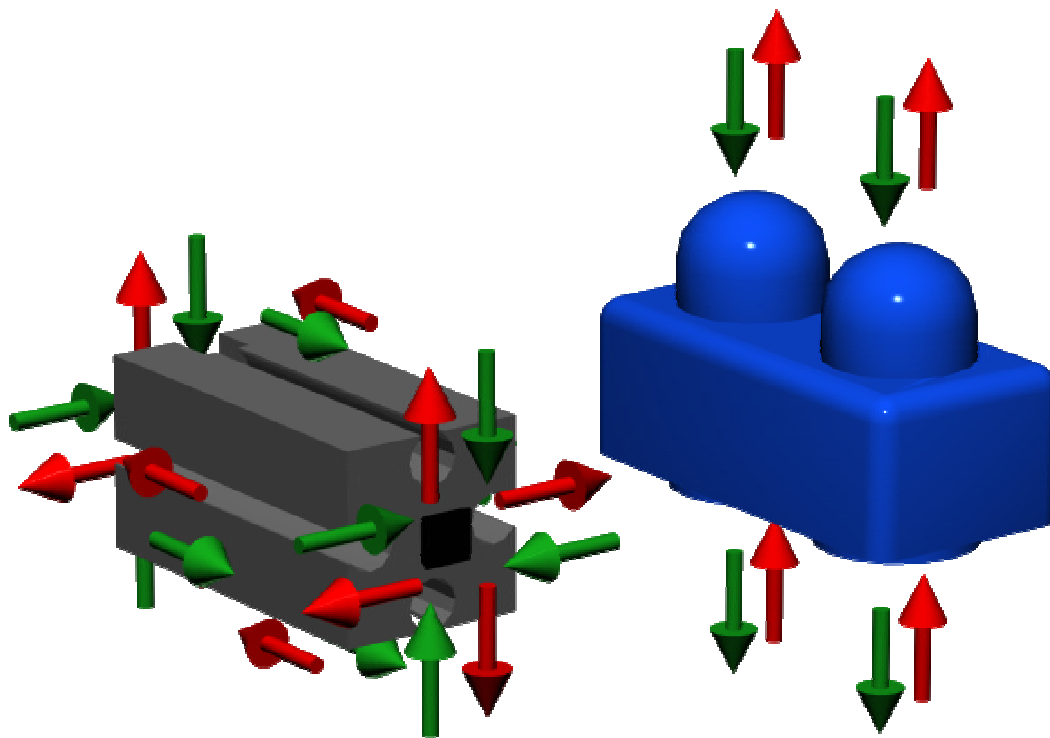


Abbildung 19: Verbindungsoptionen zweier unterschiedlicher Bausteintypen.

Dieses Prinzip legt fest, welche Bausteine zueinander passen und auf welche Weise eine mögliche Verbindung zu realisieren ist. So wie einfache Bauklötze Schicht für Schicht zu voluminösen Bauten aufeinander gestapelt werden, lassen sich andere zu untereinander verbundenen Komplexen zusammenstecken oder werden unter Zuhilfenahme von Werkzeugen fest miteinander verschraubt. Dabei setzt sich jedes dieser Systeme aus einer Reihe von Bausteintypen zusammen, die untereinander in verschiedenen Variationen kombiniert werden können und in ihrer Gesamtheit ein in sich geschlossenes Konstruktionsprinzip repräsentieren.

Aufgrund ihrer speziellen Eigenschaften bestimmen diese Konstruktionselemente nicht nur den konzeptionellen Aufbau eines Modellbaukastens, sondern legen auf diesem Wege zugleich dessen Einsatzmöglichkeiten fest. Die individuelle Formgebung der einzelnen Bausteintypen und deren Funktionsprinzip entscheiden letztendlich darüber, welche Positionen sie zueinander einnehmen können und welche funktionalen Prinzipien mit ihrer Hilfe realisiert werden können (siehe Abbildung 19). Sie bilden, als Einheit zusammengefaßt, ein homogenes Gefüge, welches die gesamte statische Funktionalität eines Baukastensystems definiert und charakterisiert.

Auch hinsichtlich der Anwendungsspezifikation kommt den Bausteintypen eine zentrale Bedeutung zu. Aufgrund der dargestellten Eigenschaften bieten sie die Möglichkeit, den Vorgang des Modellierens in seiner Komplexität erheblich einzugrenzen. Wird eine umfassende und funktionsgerechte Darstellung von Modellbaukästen im Rahmen der Simulation angestrebt, bedarf es einer adäquaten Abbildung der benötigten Elemente sowohl hinsichtlich struktureller, als auch funktionaler Kriterien. Darunter fallen insbesondere folgende drei Gesichtspunkte:

- Geometrische Charakteristika
Um eine realitätsnahe Darstellung der Konstruktionen gewährleisten zu können, müssen die virtuellen Modelle den realen Grundelementen maßstabgerecht nachempfunden werden.
- Konstruktionsrelevante Charakteristika
Die spezifischen Eigenschaften der Bausteine bezüglich des Auf- und Abbaus von Verbindungen müssen in ihrer Funktion dem Original entsprechen.
- Funktionale Charakteristika
Eigenschaften, wie z.B. mögliche Drehbewegungen von Achsen müssen in Betracht gezogen werden, um nicht nur das statische Abbild einer Originalkonstruktion simulieren zu können.

Im Rahmen eines Konstruktionssystems müssen diese Eigenschaften für jeden Bausteintyp explizit definiert werden. Es handelt sich hierbei um Objekteigenschaften, die eine konstante Informationsgrundlage des eigentlichen Modellierprozesses bilden und deshalb als eine der wichtigsten statischen Grundstrukturen zu betrachten sind.

3.3.1.2 Bausteine

Im Gegensatz zu Bausteintypen, bei denen es sich um abstrakte, „gedachte“ Konstrukte handelt, stellen Bausteine die zentralen und zugleich kleinsten *handhabbaren* Objekte eines Baukastens dar. Jeder dieser Kästen enthält eine bestimmte Anzahl solcher Bauteile, mit denen nach Belieben herumexperimentiert werden kann, die sich zu mehr oder weniger komplexen Gebilden zusammenfügen lassen und von denen immer der eine, alles entscheidende Stein fehlt.

Dieser chronische Mangel ist auf eine konkrete Abhängigkeit zwischen Bausteinen und Bausteintypen zurückzuführen. Man kann feststellen, daß jedem Bauteil ein solcher Typ eindeutig zuzuordnen ist. Letzterer bestimmt aufgrund seiner Struktur das konkrete Aussehen und die spezifische Funktionalität eines solchen Elements, d.h. die unveränderlichen Eigenschaften eines real existierenden Bausteins finden ihren Ursprung in dem dazugehörigen Bausteintypen. Auf diese Weise ist gewährleistet, daß alle Baukastenbestandteile ein zueinander kompatibles Verhalten aufweisen und somit die prinzipielle Methodik des Systems konsistent eingehalten werden kann.

Das Verhältnis zwischen Bausteintyp und Baustein läßt sich passend mit dem Konzept der objektorientierten Programmierung vergleichen. Bausteintypen definieren anhand bestimmter Eigenschaften eine Klasse konzeptioneller Objekte. Von einer solchen Klasse werden Instanzen gebildet, die eine konkrete Ausprägung dieser abstrakten Struktur darstellen. Bei greifbaren Bausteinen handelt es sich demzufolge um Instanzen der entsprechenden Bausteintyp-Klasse.

Neben dieser Beziehung besitzen Bausteine zusätzliche Eigenschaften, die sie als eigenständige Objekte auszeichnen. Diese Attribute beziehen sich im wesentlichen auf die Eigendynamik, der ein solches Element im Laufe des Modellierprozesses unterliegt. Dazu gehört sowohl dessen spezifisches Verhalten, als auch der Zustand, in dem sich der Baustein zu einem bestimmten Zeitpunkt befindet.

Diese Eigenschaften gilt es im Rahmen einer Simulation entsprechend der gegebenen Zielsetzung zu berücksichtigen und abzubilden:

- Physikalisches Verhalten

Jeder Baustein sollte einfachen physikalischen Gesetzmäßigkeiten unterliegen.

Zum Beispiel sollten Objekte, die von Benutzenden im freien Raum losgelassen werden, nicht dort verbleiben, sondern der Realität entsprechend zu Boden fallen.

Ein weiteres wichtiges Kriterium stellt die Kollisionserkennung dar. Obwohl diese Problematik bereits zu einem großen Teil durch die Positionsverfolgung der realen Bauteile gelöst ist, müssen zusätzliche Spezialfälle berücksichtigt und Ungenauigkeiten ausgeglichen werden, um eine korrekte Analyse des Konstruktionsprozesses gewährleisten zu können.

- **Lage im Raum**
Obwohl sich die räumliche Ausrichtung eines Bausteins im Laufe des Modellierprozesses häufig ändert, läßt sich dessen Position und Orientierung im Realen jederzeit auf einen Blick ausmachen. Diese offensichtlichen Objektzustände gilt es im Virtuellen durch entsprechende Attribute nachzubilden, um bei Bedarf schnell Kenntnis über den genauen Zustand einer Situation erlangen zu können.
- **Existierende Verbindungen zu anderen Bausteinen**
Der dynamische Prozeß des Auf- und Abbaus von Verbindungen zwischen Bausteinen ist einer der entscheidendsten im gesamten Verlauf eines Konstruktionsvorgangs. Entsprechend wichtig ist es darauf zu achten, welche Elemente an welcher Stelle miteinander verknüpft sind, bzw. wo offene Verbindungen das Zusammenfügen weiterer Bauteile ermöglichen. Hierzu gilt es in angemessener Zahl Informationen bereitzustellen, die eine zügige Untersuchung aller relevanten Bausteinstrukturen ermöglicht.
- **Funktionale Zustände**
Neben konstruktionsrelevanten Eigenschaften können Bausteinen auch funktionale Aufgaben zukommen. Elemente, die beispielsweise die Funktion eines Schalters inne haben, sollten sich im Rahmen einer Simulation über dessen Zustand bewußt sein und diesen entsprechend berücksichtigen.
Da im Rahmen dieser Arbeit von der Integration einer solchen Modelldynamik aus Gründen der Komplexität abgesehen wurde, bleiben entsprechenden Attribute von weiteren Betrachtungen ausgeschlossen.

3.3.1.3 Modelle

Der eigentliche Zweck von Baukastensystemen ist es, die Gestaltung unterschiedlichster Gebilde unter Zuhilfenahme bestimmter Bauteile zu ermöglichen. Im Rahmen eines festgelegten Regelwerkes wird die Möglichkeit geboten, einzelne Bausteine zu einem komplexen Modell zusammensetzen zu können. Dieses kann in seiner Gesamtheit wiederum als ein eigenständiges Objekt aufgefaßt werden und einem bestimmten Anwendungszweck dienen.

Im Zuge des eigentlichen Konstruktionsprozesses ist diese Auffassung jedoch wenig hilfreich. Sinnvoller erscheint es, nicht das zu erzeugende Endprodukt einer näheren Untersuchung zu unterziehen, sondern vielmehr die in der Phase der Entstehung befindliche Konstruktion in den Mittelpunkt des Interesses zu stellen.

Unter diesem Aspekt betrachtet, repräsentieren Modelle zunächst eine zusammengehörige Gruppe einzelner Bausteine, die untereinander in Beziehung stehen. Eine solche Einheit unterliegt, in Abhängigkeit des eingesetzten Baukastensystems, einer mehr oder minder komplexen Ordnung, die sich aufgrund der spezifischen Formation der Bausteine ergibt.

Bei einem Konstrukt dieser Art handelt es sich um eine Meta-Struktur, die im Laufe der Aufbauphase einer häufigen Wandlung unterworfen ist, da ständig neue Bestandteile hinzugefügt und andere wieder entfernt werden. Diese Eigendynamik gilt es im Rahmen der zu entwickelnden Applikation auf geeignete Weise zu reproduzieren, um die angestrebte Synchronisation zwischen realem und virtuellem Modell erzielen und aufrecht erhalten zu können. Dabei müssen bestimmte Gesetzmäßigkeiten beachtet werden:

- Jedes Modell besteht aus einer Anzahl von Bausteinen, die alle miteinander verbunden sein müssen.
- Ein Baustein, der nicht zu einem solchen Verbund gehört, stellt wiederum selbst ein eigenständiges Modell dar.
- Modelle sind sich dynamisch verändernde Konstrukte, d.h. die Anzahl der ihnen zugeordneten Elemente ist nicht konstant.
- Die Topologie eines Modells ergibt sich anhand der bestehenden Verbindungen zwischen einzelnen Bausteinen.

3.3.2 Dynamik des Konstruktionsprozesses

Anhand der im vorangegangenen Abschnitt dargestellten Objektklassifizierungen zeichnet sich bereits ab, daß der eigentliche Verlauf der Modellierung einer ganzen Reihe von Gesetzmäßigkeiten und Rahmenbedingungen unterliegt. Diese Richtlinien müssen bezüglich der zu leistenden Realitätsabbildung berücksichtigt werden, um fehlerhafte Interpretationen der aktuellen, realen Situation weitestgehend ausschließen zu können.

Ein zusätzliches Kriterium, das es in diesem Zusammenhang zu berücksichtigen gilt, ist der prozedurale Vorgang der eigentlichen Modellmontage. Wie bereits festgestellt, bestehen Modelle aus einer Anzahl von Einzelteilen, deren Zusammensetzung von den Benutzenden anhand eigener Vorstellungen bestimmt wird. Diesem Interaktionsprozeß kommt deshalb im Rahmen der Anwendungsdefinition eine zentrale Bedeutung bezüglich der zu leistenden Synchronisation zu.

Zunächst kann festgehalten werden, daß es sich bei diesem Ablauf um einen experimentellen, wenig organisierten Vorgang handelt. Dieser zeichnet sich durch das Verbinden, Trennen und Umpositionieren einzelner Elemente oder ganzer Bausteingruppen aus. Die erzielten Modellzustände der jeweiligen Entwicklungsschritte unterliegen dabei keinen festen Regeln, so daß eine Situationsanalyse immer nur in dem aktuell gültigen Kontext erfolgen kann. Anhand dieses Bezugsrahmens muß demnach der gesamte Konstruktionsprozeß nachvollzogen werden.

Bei genauerem Hinsehen zeigt sich jedoch, daß die abgeschlossenen Entwicklungsstufen keine sinnvollen Rückschlüsse auf den erfolgten Ablauf der Konstruktion zulassen. Vielmehr handelt es sich bei ihnen um genau jene Entwicklungsstadien, die es im Zuge einer Topologieanalyse zu ermitteln gilt.

Von entscheidender Relevanz hingegen ist der konkrete Übergang zwischen diesen Einzelzuständen, denn innerhalb dieses Zeitraums entscheidet sich, welchen strukturellen Veränderungen ein Modell unterworfen sein wird. Um also die korrekte Interpretation einzelner Handlungsschritte vornehmen zu können, bedarf es einer genauen Analyse aufeinanderfolgender Bearbeitungsperioden.

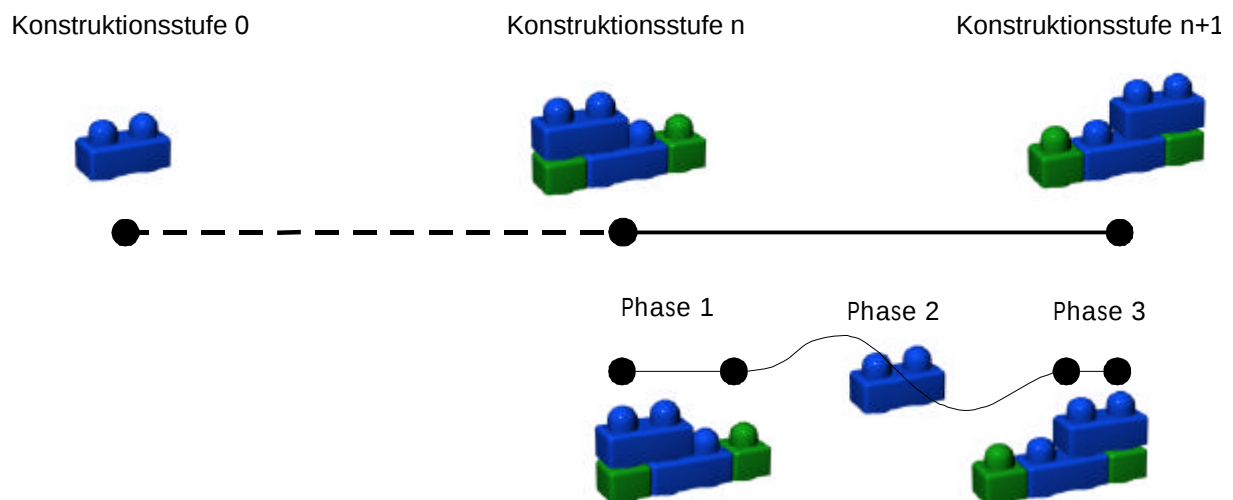


Abbildung 20: Konstruktionsstufen und -phasen des Modellierungsvorgangs.

Ein solcher Zeitabschnitt unterteilt sich wiederum in drei voneinander getrennte Phasen, anhand derer die Veränderungen an einem Modell kontrolliert nachvollzogen werden können. Auch diese Phasen ziehen sich über einen mehr oder weniger langen Zeitraum hin, so daß sich abzeichnende Entwicklungen anhand bestimmter Kriterien beurteilen lassen(siehe Abbildung 20).

3.3.2.1 Greifen

Die Phase des Greifens kennzeichnet den Beginn einer Modellmodifikation.

Aufgrund der Tatsache, daß Benutzende einen Baustein in ihre Hand nehmen, wird deutlich, daß sich dieses Element im Interesse der betreffenden Person befindet. Es ist zu vermuten, daß das Bauteil in seiner derzeitigen Lage nicht den Vorstellungen der Modellierenden entspricht und deshalb in einem nächsten Handlungsschritt seiner gegenwärtigen Konfiguration entnommen wird.

Von diesen Voraussetzungen ausgehend besteht die gerechtfertigte Grundannahme, einige vorausschauende, spekulative Betrachtungen vorzunehmen. Diese stehen konsequenter Weise

in direktem Zusammenhang mit der anstehenden Modifikation des betroffenen Modells. Ausgehend von den über einige Meßzyklen¹ hinweg registrierten Handbewegungen, werden nachstehende Untersuchungen getätigt und entsprechend umgesetzt:

- Feststellen der Bausteinlage.
Um weitere Abhängigkeiten überprüfen zu können, muß in einem ersten Schritt festgestellt werden, welcher konkrete Baustein gegriffen wurde und an welcher Position im Raum sich dieser befindet.
- Ermitteln der Modellzugehörigkeit.
Wie bereits erwähnt, ist jeder Baustein Mitglied eines ihm übergeordneten Modells. Dieses gilt es unter den bestehenden ausfindig zu machen, um feststellen zu können, welche Stellung diesem Einzelteil innerhalb der Modellstruktur zukommt. Zwei unterschiedliche Fälle und eine Ausnahmesituation gilt es zu unterscheiden:
 - Bei dem Baustein handelt es sich um das alleinige Element des Modells.
Dieser Zustand stellt einen Trivialfall dar, der keine weiteren Betrachtungen nach sich zieht. Die Phase des Greifens kann direkt an dieser Stelle beendet werden.
 - Das Modell besteht aus mehr als einem Element.
Aus dieser Situation heraus folgt zwangsläufig, daß der gegriffene Baustein mit anderen Elementen des gleichen Modells in Verbindung steht.
 - Der Baustein ist keinem Modell zuzuordnen.
Dieser Sonderfall tritt genau dann ein, wenn es sich hierbei um ein neues Element handelt, welches bisher noch nicht in den Konstruktionsprozeß integriert war. Um aus diesem Zustand wieder zu konsistenten Voraussetzungen zu gelangen, wird speziell für dieses Einzelteil ein neues Modell generiert. Aufgrund dieser Maßnahme kann dann wieder der zuerst genannte Fall greifen.

Alle nachfolgenden Betrachtungen werden nur dann berücksichtigt, wenn das ermittelte Modell mehr als einen Baustein enthält.

- Festhalten des aktuellen Modellzustands.
Diese Maßnahme muß ergriffen werden, um im Falle einer Fehlprognose (der Baustein wird unverändert wieder losgelassen) die ursprünglich vorgefundene Ausgangslage reproduzieren zu können.
- Ermitteln der Bewegungsrichtung.
Die direkt nach dem Greifen festgestellten Positionsänderungen werden auf das betreffende Modell übertragen. Zusätzlich wird anhand dieser Werte die Richtung bestimmt, in die der gegriffene Baustein von den Benutzenden geführt wird.

¹ Die Bewegung der Hand wird mit Hilfe eines Tracking-Systems realisiert, das Meßwerte in einer Frequenz von 60 Hz liefert.

- Heraustrennen des Bausteins aus dem Modell.
Unter Berücksichtigung der zuvor berechneten Bewegungsrichtung und der dem Baustein zugrundeliegenden Verbindungseigenschaften (hier tritt der entsprechende Bausteintyp in Erscheinung) sind bestehende Verknüpfungen zu angrenzenden Bauteilen aufzuheben. Dieser Schritt ist an alle Modellelemente weiterzuleiten, die sich nicht direkt oder indirekt von dem gegriffenen Baustein trennen lassen.
Als Ergebnis erhält man ein neues Modell mit genau jenen Bausteinen, die sich unter den gegebenen Voraussetzungen aus der ursprünglichen Konstruktion entfernen ließen.
- Kontrolle der verbleibenden Restmenge.
Konnte in dem vorangegangenen Schritt eine Aufteilung des Modells vorgenommen werden, bleibt nach erfolgter Trennung eine Anzahl von Bauelementen zurück. Für diese Gruppe muß überprüft werden, ob es sich nach wie vor um einen einzelnen Verbund handelt, oder ob sich aufgrund der entnommenen Bausteine mehrere kleinere Modelle gebildet haben.
Zusätzlich müssen diese Modelle in ihre ursprüngliche Ausgangslage zurückversetzt werden, um Inkonsistenzen zur Realität zu verhindern.

Nach Abschluß dieser Liste von Untersuchungen und der Anwendung entsprechender Maßnahmen endet an dieser Stelle die Phase des Greifens und wird direkt von der Bewegungsphase abgelöst.

3.3.2.2 Bewegen

Im Zuge der Bewegungsphase verändert sich die Lage des gegriffenen Modells kontinuierlich.

Dieser dynamische Prozeß wird im Realen aufgrund der vollzogenen Handbewegungen der Benutzenden kontrolliert und gesteuert. Als primäres Ziel der Akteure steht dabei die Neupositionierung des gegriffenen Modells im Kontext der Gesamtkonstruktion im Vordergrund.

Aus anwendungsspezifischer Sicht ist zunächst das kontinuierliche Erfassen des Bewegungsablaufes in den Mittelpunkt des Interesses zu stellen. Die einzelnen Bausteine müssen in Abhängigkeit zu dem gegriffenen Element stetig an dem sich verändernden Realzustand ausgerichtet werden. Dies ist die generelle Voraussetzung, um neue Formationen innerhalb der gegebenen Modellkonstruktion korrekt erkennen zu können.

Die Bewegungsphase endet schlagartig mit dem Loslassen des entsprechenden Elements. Da von diesem Augenblick an keine weiteren Veränderungen des gegriffenen Modells mehr stattfinden, müssen alle für diesen Fall relevanten Informationen bereits vorab ermittelt worden sein.

Als weiteres Problem kommt hinzu, daß sich der entscheidende Moment des Loslassens nicht im Voraus erkennen läßt. Das unbestimmte Auftreten dieses Ereignisses führt dazu, daß in jeder neu auftretenden Konstellation das Ende der Bewegungsphase mit in Betracht gezogen werden muß.

Folgendes gilt es im Zuge der Bewegungsphase zu prüfen und zu berücksichtigen:

- Feststellen der Lage des gegriffenen Bausteins.
- Aktualisieren der Modellposition und Orientierung.
Eine neue räumliche Ausrichtung des gegriffenen Bausteins muß auf die verbleibenden Teile des Modells übertragen werden, da es sich hierbei um einen zusammengehörigen Komplex handelt. Das gegriffene Element dient dabei als lokaler Bezugspunkt relativer Transformationen.
- Ermitteln der Bewegungsrichtung.
Wie bereits in der ersten Phase wird auch hier die derzeitige Richtung bestimmt, in die das Modell von den Modellierenden geführt wird.
- Identifizieren freier Anknüpfungspunkte.
Im Rahmen der Modellstruktur müssen jene Bausteine ausgewiesen werden, die aufgrund offener Verbindungsstellen die Fähigkeit besitzen, das gegriffene Modell an ein weiteres zu koppeln.
- Feststellen potentieller Kombinationsansätze.
Die im vorangegangenen Schritt ermittelten Anknüpfungspunkte müssen dahingehend untersucht werden, ob ihre freien Ansätze auch unter Berücksichtigung der Bewegungsrichtung eine Verbindung zu anderen Bausteinen erlauben.
Die nach dieser Überprüfung verbleibende Anzahl Bausteine repräsentiert bezüglich der gegebenen Lage eine Schnittstelle zur Ankopplung weiterer Modelle. Die ausgewählten Elemente repräsentieren demnach das gegenwärtige Potential des Modells, neue Verbindungsstrukturen etablieren zu können.

Die hier dargestellte Folge von Berechnungen wird zyklisch, nach jeder Lageveränderung wiederholt durchlaufen, bis die Bewegungsphase durch das Loslassen des gegriffenen Bausteins terminiert wird.

3.3.2.3 Loslassen

Die Phase des Loslassens realisiert das Zusammenfügen von Modellen.

Nachdem die Benutzenden ihren gegriffenen Baustein losgelassen haben, tritt in der Realität sofort ein statischer Modellzustand ein. Der Modifikationsprozeß ist mit dieser Handlung abgeschlossen, da keine Veränderungen der Konstruktion mehr möglich sind.

Im Rahmen der Simulation ist dieser konstante Zustand jedoch noch nicht erreicht, denn das bislang von Veränderungen betroffene Modell hat als eigenständiges Objekt nach wie vor Bestand.

In dieser Phase gilt es deshalb die Aufgabe zu lösen, die verbleibende Bausteingruppe wieder korrekt in den Gesamtzusammenhang der Konstruktion zu integrieren. Dieser Vorgang, der in der Realität automatisch im Zuge der Modellmanipulation vonstatten geht, beansprucht im Virtuellen einen zusätzlichen Bearbeitungszyklus. Dabei wird intensiv auf die Informationsbasis der vorangegangenen Bewegungsphase zurückgegriffen.

Das Zusammenfügen von Modellen gliedert sich in folgende Beobachtungen und Teilschritte:

- Betrachtung der potentiellen Verbindungsbausteine.
Anhand der ausgewählten Bausteine, die eine Verbindung zu anderen erlauben, kann festgestellt werden, ob weitere Analysen innerhalb dieser Phase nötig sind.
 - Es gibt keine Elementauswahl.
Kein Baustein ist in der Lage, eine neue Verknüpfung zu realisieren, so daß das Modell in seinem bisherigen Zustand verbleibt.
 - Mindestens ein Baustein ist ausgewählt.
Prinzipiell besteht die Möglichkeit, das Modell mit anderen zu verbinden. In diesem Fall bleibt festzustellen, ob ein passendes Zielmodell existiert.
- Auswahl möglicher Zielmodelle.
Aus der Menge der unbeweglichen Modelle sind solche herauszufiltern, die sich in unmittelbarer Nähe zu dem gegriffenen Modell befinden.
Können diese nicht ausfindig gemacht werden, bleibt das Modell auch in diesem Fall in seinem ursprünglichen Zustand bestehen.
- Bestimmen passender Zielbausteine.
Analog zur Auswahl der potentiellen Anknüpfungspunkte des gegriffenen Modells, werden entsprechende Bausteine der Zielmodelle identifiziert. Diese müssen grundsätzlich eine freie Verbindungsmöglichkeit aufweisen. Zusätzlich ist zu beachten, daß die offenen Ansatzstellen in der Lage sind, sich in entgegengesetzter Bewegungsrichtung¹ mit einem in der Nähe befindlichen Baustein des gegriffenen Modells zu verknüpfen.
Nur wenn dieser Umstand gegeben ist, können die zwei betrachteten Bausteingruppen miteinander gekoppelt und somit in ein einzelnes Modell überführt werden.

¹ Das Umkehren der Bewegungsrichtung kommt einem Perspektivwechsel gleich, bei dem das aktuell untersuchte Modell quasi in die Rolle des gegriffenen versetzt wird und sich aus dieser Sicht darauf zubewegt.

- Absetzen des Modells.

In den Fällen, in denen das Modell nicht an ein weiteres gekoppelt werden kann, wird davon ausgegangen, daß der Konstrukteur dieses abgestellt oder fallengelassen hat. Entsprechend erfolgt eine Ausrichtung des Modells auf der Arbeitsoberfläche.

Mit dem Abschluß dieser letzten Bearbeitungsphase ist ebenso der Übergang von einem zulässigen Modellzustand zum nächsten beendet. Sofern keine Fehlinterpretationen im nachvollzogenen Handlungsablauf (z.B. durch auftretende Abweichungen bei der Positionsmessung der Hand) aufgetreten sind, befinden sich das reale und virtuelle Modell in einem adäquaten Zustand. Da zu diesem Zeitpunkt keine Veränderungen mehr innerhalb der Modellkonstruktion möglich sind, verfällt der Simulationsprozeß in einen inaktiven Modus. Erst wenn erneut ein Baustein von den Anwendenden aufgenommen wird, muß ein nächster Iterationsschritt angestoßen werden, um erneut den anstehenden Konstruktionsvorgang auf das virtuelle Modell zu übertragen.

Kapitel 4

Spezifikationen

Nachdem sich anhand der dargestellten Problem- und Strukturanalyse gezeigt hat, daß es prinzipiell möglich sein sollte, das angestrebte Ziel der synchronen Modellierung in die Praxis umzusetzen, werden in dem nun folgenden Kapitel die gewonnen Erkenntnisse in geeignete Datenstrukturen und Algorithmen umgesetzt. Dabei wird in Hinblick auf die Implementierung von einem objektorientierten Ansatz ausgegangen.

Neben der reinen Methodik, die es im Zuge implementierungsrelevanter Überlegungen zu berücksichtigen gilt, muß den Aspekten der Modellkonsistenz und Verarbeitungsgeschwindigkeit eine gewisse Aufmerksamkeit gewidmet werden.

Konsistenz meint in diesem Zusammenhang, daß Toleranzmechanismen vorgesehen werden, die auftretende Ungenauigkeiten in einem gewissen Rahmen abzufangen vermögen. Geschieht dies nicht, besteht die Gefahr, daß zum Beispiel das Verbinden zweier Bausteine vom System nicht erkannt wird, und deshalb eine korrekte Abbildung der realen Situation nicht mehr möglich ist.

Da die Simulation des Konstruktionsprozesses synchron zur Realität erfolgen muß, ist ebenso auf eine angemessene Verarbeitungsgeschwindigkeit zu achten. Die Anwender sollten in der Lage sein, ihre normale Arbeitsgeschwindigkeit beibehalten zu können, ohne daß es dabei zu erheblichen Unterbrechungen durch das System kommt.

4.1 Datenstrukturen

Die Spezifikation der Datenstrukturen orientiert sich maßgeblich an den Komponenten, die sich bereits im Rahmen des Analyseprozesses als eigenständige Objektgruppierungen herauskristallisiert haben. Diese werden nach Bedarf gewissen strukturellen Erweiterungen, Abwandlungen oder Ergänzungen unterzogen, um auf geeignete Weise in die Gesamtstruktur der entwickelten Spezifikation integriert werden zu können.

Darüber hinaus werden gänzlich neue, bisher nicht berücksichtigte Objektklassen eingeführt. Deren Zweckmäßigkeit hat sich erst im Zuge der Entwicklung konkreter Algorithmen und Verfahrensweisen ergeben oder steht in direktem Zusammenhang zur Implementierung.

Bevor im weiteren Verlauf dieses Abschnitts die Spezifikation der einzelnen Klassen und deren Funktion im Gesamtkontext der Anwendung erläutert werden, soll zunächst noch auf ein grundsätzliches Unterscheidungsmerkmal hingewiesen werden.

Jedes hier vorgestellte Objekt kann einer von zwei sich gegenseitig ausschließenden Gruppierungen zugeordnet werden. Das wesentliche Unterscheidungsmerkmal dieser Kategorien stellt deren Grad an Abstraktion im Sinne der Abbildung von Welt dar. Während sich zu den Objekten der einen Gruppe immer auch ein direkter oder indirekter Bezug zum Realen herstellen läßt, handelt es sich anderenfalls um rein fiktive Konstrukte, denen ausschließlich im virtuellen Kontext der Simulation eine spezielle Bedeutung zukommt.

- Konzeptionelle Objekte

Konzeptionelle Objekte definieren abstrakte Strukturen.

Diese Strukturen sind für den Aufbau eines Baukastensystems von zentraler Bedeutung, stehen darüber hinaus aber nicht in unmittelbarem Zusammenhang mit dem eigentlichen Vorgang der Modellkonstruktion. Ihre primäre Aufgabe besteht darin, das funktionale *Konzept* eines Baukastensystems festzulegen und im Rahmen des Konstruktionsprozesses verfügbar zu machen. Konzeptionelle Objekte legen Relationen und Verhaltensweisen fest, die in der Realität durch gegebene, natürliche Umstände von vorn herein Bestand haben.

- Repräsentative Objekte

Repräsentative Objekte besitzen einen Bezug zum Gegenständlichen.

In den Konstruktionsprozeß mit Bausteinen sind eine Menge von Gegenständen, Personen, Räumlichkeiten und Prozesse involviert, die im Zuge der Simulation *repräsentiert* werden müssen. Entsprechend werden Strukturen benötigt, die diese objektiv vorgefundenen Gegebenheiten adäquat abzubilden vermögen.

Die Darstellung der einzelnen Klassen beginnt mit der Erläuterung der konzeptionellen Objekte, gefolgt von den repräsentativen. Innerhalb dieser zwei Kategorien wird eine Beschreibung entsprechend des logischen Systemaufbaus nach dem Top-Down Prinzip vorgenommen.

4.1.1 Baukastensysteme

Innerhalb der Baukastensystem-Klasse wird die grundlegende, statische Funktionalität eines jeden Baukastens definiert und zusammengefaßt. Jeder einzelne dieser Baukästen setzt sich aus einer Reihe von Einzelkomponenten - den Bausteintypen - zusammen, die in ihrer Gesamtheit dessen eigentliches Konstruktionspotential bestimmen.

Da es sich bei einem solchen System um ein konzeptionelles Objekt handelt, sind dessen Strukturen nicht unmittelbar an einen konkreten Modellervorgang gebunden. Vielmehr besteht eine umgekehrte Abhängigkeit zwischen Konstruktionsprozeß und Baukastensystem, denn der konkrete Vorgang des Konstruierens ist ohne detaillierte Kenntnisse über die grundlegenden funktionalen Charakteristika der beteiligten Bausteine nicht nachvollziehbar. Hinsichtlich der Implementierung ergeben sich im Wesentlichen folgende Anforderungen an eine Baukastensystem-Klasse:

- **Persistenz:**
Aufgrund der potentiellen Eigenständigkeit von Baukastensystemen erscheint es sinnvoll, diese unabhängig vom Konstruktionsprozeß in separate Dateien abspeichern zu können.
- **Variabler Umfang:**
Da sich unterschiedliche Baukastensysteme auch aus einer unterschiedlich großen Anzahl von Bausteintypen zusammensetzen, kann dessen Umfang nicht auf eine konstante Größe festgelegt werden.
- **Dedizierter Zugriff auf einzelne Elemente:**
Da im Zuge des Konstruktionsprozessen häufig auf die einzelnen Elementtypen eines Baukastensystems zurückgegriffen wird, bedarf es hierfür geeigneter Zugangsmechanismen.

Infolge dieser Anforderungen bietet es sich an, Baukastensysteme in Gestalt einer Hashing-Tabelle (siehe [Sed1992], S. 231) zu realisieren. Dabei ist zu berücksichtigen, daß jedes darin enthaltene Element anhand eines Schlüsselattributes eindeutig identifizierbar sein muß.

4.1.2 Elementtypen

Elementtypen stellen die programmtechnischen Pendanten zu den bereits bekannten Bausteintypen dar. Entsprechend müssen deren genannte Charakteristika (siehe Kapitel 3.3.1.1) in eine adäquate Objektstruktur überführt werden.

Folgende Aspekte gilt es in die Klasse der Elementtypen zu integrieren:

- **Geometrie:**
Der Geometrie eines Bausteintypen kommt eine zentrale Bedeutung im Rahmen dieser Anwendung zu. Da die Entwicklung des Prototypen jedoch unter Zuhilfenahme einer Programmbibliothek (WorldToolKit) zur Visualisierung dreidimensionaler Szenen erfolgt, bedarf es keiner eigenständigen Implementierung dieser aufwendigen Strukturen. Statt dessen wird davon ausgegangen, daß die benötigten geometrischen Informationen bereits in externen Dateien entsprechenden Formats vorliegen. Von den Anwendern ist deshalb vorab, unter Zuhilfenahme geeigneter Modelliertools, ein virtuelles Abbild der Bausteine

zu erstellen. Diese Geometriedateien werden dann entsprechend innerhalb der Klasse referenziert und verwaltet.

- **Dimensionierung:**
Obwohl die eigentlich darstellungsrelevanten Informationen aus externer Quelle bezogen werden, erscheint es sinnvoll, die Dimensionierung dieser Körper variabel zu gestalten. Dadurch wird es möglich, eine Anpassung der Objektgröße vornehmen zu können, um auf diese Weise eine maßstabgerechte Abbildung des Originals zu erhalten. Innerhalb der Klasse werden die Höhe, Breite und Länge, als auch der lokale Ursprung des Objektes als eigenständige Attribute festgehalten.
- **Verbindungseigenschaften:**
Bausteintypen besitzen eine unterschiedliche Anzahl individuell ausgeprägter Anknüpfungspunkte, die jeweils eine mögliche Verknüpfung zu anderen Bausteinen charakterisieren. Im Zusammenhang mit dem Entwurf verbindungsspezifischer Algorithmen wurde festgestellt, daß es sich hierbei um relativ komplexe Konstrukte handelt. Es wurde deshalb entschieden, diese innerhalb einer eigenständigen Klasse zu kapseln.
Trotz dieser Auslagerung muß die Anzahl der Verbindungspunkte des Elementtypen festgehalten werden und ein schneller Zugriff darauf gewährleistet sein. Dazu bietet sich auch in diesem Falle die Verwendung einer Hashing-Tabelle an (siehe Kapitel 4.1.1).

4.1.3 Anknüpfungspunkte

Anknüpfungspunkte definieren die bestehenden Verbindungsmöglichkeiten von Elementtypen. Da diese Funktionalität unterschiedlichen Ausprägungen unterliegt, ist jeder Anknüpfungspunkt immer nur seinem übergeordneten Elementtypen zuzuordnen und damit innerhalb eines Baukastensystems eindeutig identifizierbar.

Im Rahmen des zu entwickelnden Prototypen wurden speziell solche Verbindungen in Betracht gezogen, die auf einem Steckprinzip beruhen. Dies ist zum Beispiel bei allen Arten von LEGO-Steinen und der Mehrzahl von Fischertechnik-Bausteinen der Fall. Schraubverbindungen sind mit Hilfe dieser Struktur hingegen nicht abzubilden, da immer nur zwei Komponenten an einer Verknüpfung beteiligt sein können.

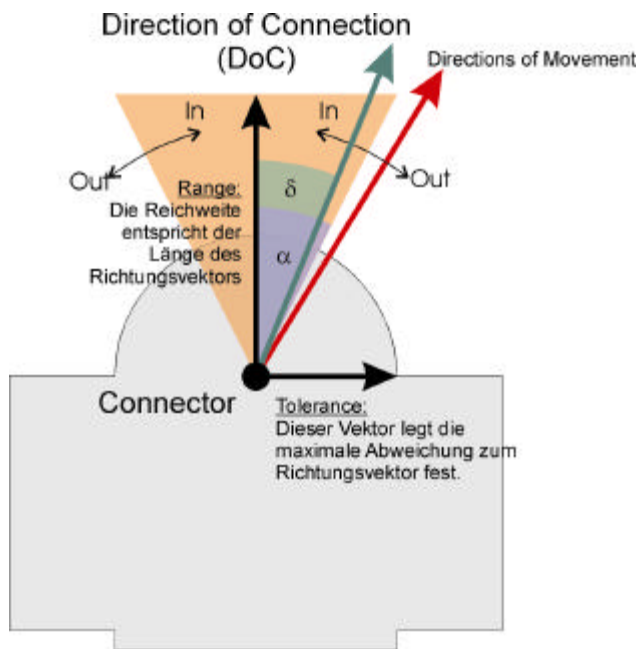


Abbildung 23:
Grafisch dargestellte Verbindungslogik eines Anknüpfungspunktes.

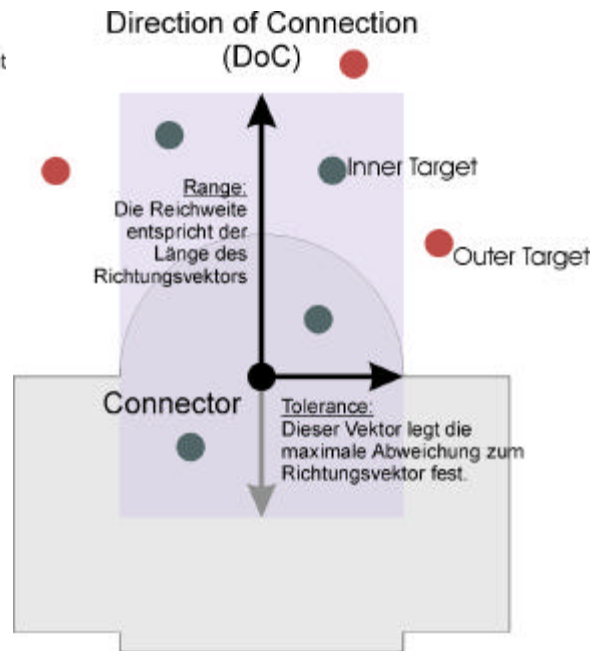


Abbildung 24:
Grafische Repräsentation des Nähe-Kriteriums eines Anknüpfungspunktes.

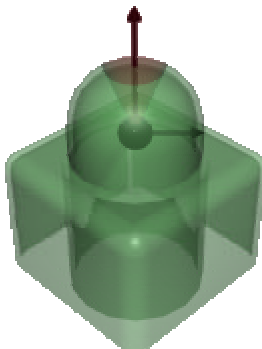


Abbildung 22: 3D-Ansicht der Verbindungslogik.



Abbildung 21: 3D-Ansicht des Nähe-Kriteriums.

Ein Anknüpfungspunkt setzt sich insgesamt aus vier, teilweise voneinander abhängigen Parametern zusammen. Anhand dieser kann der Prozeß des Zusammenbauens und Trennens zweier Bausteintypen entsprechend realitätsnah abgebildet werden (siehe Abbildung 21/24 und 22/23):

- **Position:**
Anhand der Position wird festgelegt, an welcher Stelle des Bausteintypen ein weiteres Element angefügt werden kann. Die räumliche Lage des Verbindungspunktes bezieht sich dabei auf den lokalen Ursprung des Elementtypen und ist frei definierbar. Er kann sich

zum Beispiel auch außerhalb des Bausteins befinden, wie es im Falle eines einfachen Fischertechnik-Elements nötig wäre.

- Verbindungsrichtung:

Die Verbindungsrichtung (Direction of Connection / DoC) steht in engem Zusammenhang mit der Bewegungsrichtung des Elementtypen. Anhand dieses Richtungsvektors wird bestimmt, unter welchen Bedingungen sich der Zustand einer Verbindung ändern kann (Siehe Kapitel 4.1.7). Stimmt die Bewegungsrichtung eines Bausteins mit diesem Vektor weitestgehend überein, besteht an dieser Stelle die potentielle Möglichkeit eines Verknüpfungsaufbaus. Bei einem Verbindungsabbau ist hingegen von der umgekehrten Bewegungsrichtung auszugehen.

- Toleranz:

Der Toleranzwert eines Anknüpfungspunktes wird in zweierlei Hinsicht benötigt. In Bezug auf bestehende Verbindungspotentiale legt er die maximal zulässige Abweichung der Bewegungsrichtung von der Verbindungsrichtung fest. Konkret handelt es sich hierbei um einen Winkel α , dessen Wert sich wie folgt ermittelt lässt (siehe Abbildung 23):

$$\alpha = \sin^{-1}\left(\frac{t}{r}\right)$$

Darüber hinaus wird dieser Wert mit herangezogen, um das Kriterium der Nähe festzusetzen (siehe Abbildung 24). Anschaulich dargelegt handelt es sich bei diesem Attribut um einen Zylinder, der sich um den Anknüpfungspunkt herum aufspannt und auf diese Weise einen Nahbereich spezifiziert.

- Reichweite:

Die Reichweite entspricht der Länge des Vektors, der die Verbindungsrichtung festlegt:

$$r = \left| \overrightarrow{DoC} \right|$$

Wie schon der Toleranzwert, fließt auch dieser sowohl mit in die Berechnung der maximalen Abweichung der Bewegungsrichtung, als auch mit in die Bestimmung des Nahbereiches ein (siehe Abbildung).

Obwohl der hier dargestellte Aufbau der Anknüpfungspunkte möglicherweise einige Fragen offen lässt, soll an dieser Stelle auf eine weiterführende Beschreibung der einzelnen Attribute verzichtet werden. Eine umfassende Diskussion der funktionalen Aspekte erfolgt im Zusammenhang mit der Erläuterung der Algorithmen.

Mit dem Ende dieses Abschnittes wird gleichzeitig der Bereich der konzeptionellen Objekte verlassen. Die bisher vorgestellten Klassen stellen eine hinreichend breite Informationsgrundlage dar, auf die im Rahmen der Modellkonstruktion aufgesetzt werden kann. Entsprechend beziehen sich alle nachfolgenden Ausführungen auf die Definition repräsentativer Objekte.

4.1.4 Die Szene

Oft gibt es in der Realität Orte, die speziell auf die dort ausgeübten Betätigungsfelder zugeschnitten sind. Solche Arbeitsbereiche zeichnen sich durch ein spezielles Umfeld aus, in dem die handelnden Personen in ihren Tätigkeiten entsprechend unterstützt werden. Für den Fall der Modellkonstruktion würde in einem solchen Bereich zum Beispiel eine großzügige Arbeitsfläche vorzufinden sein, um dort die geplanten Modellkonstruktionen aufbauen zu können.

Dieser Räumlichkeit entsprechend gibt es auch im Rahmen der Modellsimulation ein Objekt, unter dem alle für den Konstruktionsprozeß relevanten Strukturen zusammengefaßt werden. Primäre Aufgabe dieses Objektes ist es, den Verlauf der Konstruktion zu koordinieren, aktuell bestehende Modelle zu verwalten und diese entsprechend der erfolgten Interaktionen zu verändern. Aufgrund dieser umfassenden Aufgabe sind folgende Eigenschaften eines Szenen-Objektes erforderlich:

- **Dynamische Modellverwaltung:**
Da innerhalb einer Szene alle darin befindlichen Modellkonstruktionen zusammengefaßt werden, müssen diese auch auf eine geeignete Weise gehandhabt werden. Aufgrund der sich ständig verändernden Szenenstruktur ist es notwendig, an diese Dynamik angepaßte Mechanismen zur Modellverwaltung einzusetzen.
Aufgrund ihres häufigen Wandels sind die einzelnen Modelle jedoch nicht durch ein eindeutiges Attribut, wie zum Beispiel einen Namen, ausgezeichnet (siehe Kapitel 4.1.5). Aus diesem Grund entfällt hier die Verwendungsmöglichkeit einer Hashing-Tabelle. Statt dessen werden die Objekte innerhalb einer einfachen, doppelt verketteten Liste gespeichert und anhand eines explizit zu benennenden Bausteins identifiziert.
- **Zustandserfassung:**
Die Szene muß über den aktuellen Zustand des Konstruktionsprozesses informiert sein. Im Rahmen der Simulation ist es die Aufgabe des Szenen-Objekts, die Übergangsphasen von einem Konstruktionszustand zum nächsten zu verfolgen und entsprechende Reaktionen einzuleiten.
- **Umfassende Topologieanalyse:**
Innerhalb der einzelnen Bearbeitungsphasen muß der Verlauf der Modellierung nachvollzogen und analysiert werden. Dazu ist es unter anderem Voraussetzung, daß das

aktuell gegriffene Modell im Kontext der Szene bekannt ist und sich von den verbleibenden Bausteingruppen entsprechend unterscheiden läßt. Es wird deshalb als ein ausgezeichnetes Objekt gesondert gespeichert.

- Persistenz:
Aufgrund dessen, daß alle relevanten Datenstrukturen innerhalb einer Szene zusammengefaßt sind, muß an dieser Stelle auch die Möglichkeit geboten werden, das Ergebnis einer Konstruktion als Datei abzuspeichern.

4.1.5 Modelle

Hierbei handelt es sich um die simulationsspezifischen Gegenstücke realer Modellkonstruktionen. Wie bereits im Rahmen der Anwendungsanalyse festgestellt wurde (siehe Kapitel 3.3.1.3), fassen Modelle eine Gruppe von Elementen zusammen, die alle untereinander verbunden sein müssen. Wird ein Element aus diesem Verbund gelöst, entsteht zwangsläufig eine Anzahl neuer Modelle (mindestens zwei). Entsprechend unterliegen Modelle im Rahmen des Konstruktionsprozesses einer Eigendynamik, der im Zuge der Implementierung entsprochen werden muß:

- Variable Elementanzahl:
Aufgrund der häufig wechselnden Anzahl von Bausteinelementen ist es nicht möglich, den Umfang eines Modells auf eine feste Größe zu beschränken. Sowohl eine Verminderung, als auch einen Zuwachs an Elementen gilt es an dieser Stelle zu berücksichtigen.
- Schneller Zugriff auf einzelne Elemente:
Aufgrund der Veränderungen, denen ein Modell im Laufe des Konstruktionsprozesses unterliegt, wird auf eine eindeutige Namensgebung verzichtet. Es ist ausschließlich anhand seiner Bestandteile, den Bausteinelementen identifizierbar. Um einen direkten Zugriff auf diese Bausteine zu erlauben, wird auch hier auf die Anwendung einer Hashing-Tabelle zurückgegriffen (Siehe Kapitel 4.1.1).
- Kennzeichnen potentieller Verbindungsmöglichkeiten:
Die im Zuge der Bewegungsphase ermittelte Auswahl möglicher Verbindungsbausteine muß über den Zeitraum mehrerer Bearbeitungszyklen hinweg aufrecht erhalten werden. Ausgehend von der direkten Zugriffsmöglichkeit auf einzelne Modellkomponenten, reicht es in diesem Falle aus, die entsprechenden Schlüsselattribute in einer Liste zwischen zu speichern.

Wie die Betrachtung des Konstruktionsablaufes gezeigt hat, kann immer nur ein ausgezeichnetes Modell zur Zeit bewegt und manipuliert werden. Dieses gegriffene Modell gilt es

im Rahmen dieser Spezifikation gesondert zu betrachten. Es handelt sich hierbei um ein von der Modell-Klasse abgeleitetes Objekt mit folgenden zusätzlichen Eigenschaften:

- Herausstellen des gegriffenen Bausteins
Das entsprechende Element wird ausdrücklich referenziert.
- Bestimmen der Bewegungsbahn:
Obwohl die Berechnung der Bewegungsrichtung innerhalb einer eigenen Klasse gekapselt wurde, erscheint mir deren Darstellung in diesem Abschnitt der Arbeit unangebracht. Statt dessen erfolgt die Erläuterung der Richtungsbestimmung im Rahmen der Darstellung relevanter Algorithmen, da es sich hierbei um den Kontrollmechanismus eines fortschreitenden Prozesses handelt (Siehe Kapitel 4.2.1)

4.1.6 Elemente

Elemente repräsentieren real existierende Bausteine. Entsprechend müssen deren im Abschnitt 3.3.1.2 erläuterten Attribute in eine geeignete Objektstruktur umgesetzt werden:

- Schlüsselattribut:
Innerhalb einer Szene ist jedes Element eindeutig identifizierbar. Entsprechend besitzt es ein Schlüsselattribut, welches über den Zeitraum des Konstruktionsprozesses unverändert bestand hat.
Diese Konsistenz wirkt sich in Hinblick auf die Modellzugehörigkeit des Elements in zweierlei Hinsicht aus. Da auch bei einem Wechsel des Besitzermodells die Bausteinbezeichnung erhalten bleibt, besteht einerseits die Möglichkeit, modellübergreifende Beziehungen zwischen Elementen zu erfassen. Andererseits ist es längerfristig nicht praktikabel, Modelle anhand eines Bausteinbezeichners zu spezifizieren, da dieser bereits Bestandteil eines anderen Modells sein könnte.
- Basistyp:
Da die statische Informationsbasis eines Bausteinelements innerhalb eines spezifischen Elementtypen gekapselt ist, muß im Rahmen der Simulation eine permanente Zugriffsmöglichkeit auf diesen Basistyp bestehen. Entsprochen wird dieser Forderung in Form eines Objektverweises.
- Lage im Raum:
Die Position und Orientierung eines Bausteinelements sind von fundamentaler Bedeutung für den Gestaltungsvorgang und werden deshalb für jeden Baustein gesondert festgehalten.
- Modellzugehörigkeit:
Wie bereits dargelegt, ist jedes Element einer Szene gleichzeitig Bestandteil eines Modells. Da sich letzteres nicht anhand eines Schlüsselattributes auszeichnet, muß die

Modellzugehörigkeit von den Elementen selbst verwaltet werden. Für diesen Zweck wird innerhalb der Klasse ein entsprechender Objektverweis vorgesehen.

- Verknüpfungszustände:

An dieser Stelle wird die bereits begonnene Auslagerungsstrategie von Verbindungsstrukturen konsequent fortgesetzt. So wie es schon bei den Anknüpfungspunkten geschehen ist, werden auch die Verknüpfungszustände der einzelnen Bausteine an eine eigene Objektklasse gebunden. Da zu jedem Anknüpfungspunkt eine Verbindungsausprägung existiert und sich die Struktur eines Baukastensystems nicht unerwartet ändert, ist für jedes Bausteinelement eine fest vorgegebene Anzahl von Verbindungsattributen vorzusehen. Es bietet sich daher an, die Verbindungsobjekte in einem Feld fester Größe abzulegen.

4.1.7 Verbindungszustände

Als fester Bestandteil eines Bausteinelements erfassen Verbindungszustände die Verknüpfungsverhältnisse eines Elements. Dabei wird auf die Eigenschaften des zugrundeliegenden Anknüpfungspunktes zurückgegriffen. Insofern repräsentieren Verbindungsobjekte keine gegenständlichen, sondern funktionale Strukturen des realen Konstruktionszustands. Unter diesem Aspekt betrachtet, muß ein solches Objekt folgende Eigenschaften aufweisen:

- Basistyp:

Der grundlegende statische Aufbau einer Verbindung ist innerhalb eines Anknüpfungspunktes definiert. Um auf diese Struktur bei Bedarf zugreifen zu können, besteht innerhalb des Verknüpfungsobjektes ein Verweis auf das entsprechende Basisobjekt.

- Zustand:

Eine Verbindung kann grundsätzlich zwei gegensätzliche Zustände aufweisen. Bei einer bestehenden Verknüpfung befindet sich das Objekt in einem geschlossenen Zustand, anderenfalls ist die Verbindung offen. Diese Eigenschaft gilt es innerhalb der Klasse geeignet zu erfassen und zu verwalten.

- Bausteinzugehörigkeit:

Da eine Verknüpfung ohne Bezug zu einem konkreten Baustein keinen Bestand hat, besteht innerhalb der Klasse eine Referenz auf diesen Besitzerbaustein.

- Zielverbindungen:

Im Zusammenhang mit den Verarbeitungsphasen des Auf- und Abbaus von Verbindungen ist es notwendig, Kenntnis über passende Partnerverbindungen zu erlangen. Dabei wird auf eine Kombination aus Zielbaustein und dessen Verbindungsstelle zurückgegriffen. Die betreffende Auswahl wird in Form einer Liste festgehalten.

Die in diesem Abschnitt dargestellten Datenstrukturen stellen zusammengefaßt die informelle Basis des Konstruktionsprozesses dar. Auf dieser Gesamtstruktur aufbauend kommen

Verfahren zum Einsatz, mit deren Hilfe der eigentliche Simulationsanteil der Anwendung abgedeckt wird. Eine Beschreibung dieser funktionalen Aspekte folgt im nächsten Abschnitt dieses Kapitels.

4.2 Algorithmen

Wie sich anhand der Anwendungsanalyse gezeigt hat, lassen sich drei grundsätzliche Phasen innerhalb eines einzelnen Konstruktionsschrittes unterscheiden. Diese Einzelschritte beschränken sich jeweils auf einen konkreten Zeitabschnitt des stattfindenden Ablaufs.

Um innerhalb dieser Phasen simulationsrelevante Veränderungen der Szenenstruktur ableiten und umsetzen zu können, muß anhand bestimmter Gesichtspunkte eine Analyse und Bewertung des gegebenen Gesamtbildes der Szene vorgenommen werden. Dieser Vorgang einer Topologieanalyse wird unter Zuhilfenahme verschiedener Algorithmen realisiert.

4.2.1 Vorhersage der Bewegungsrichtung

Ein ausgesprochen wichtiger Faktor bei der Rekonstruktion des Modellierprozessen stellt der Bewegungsverlauf eines gegriffenen Modells dar. Anhand dieser Dynamik ist zu untersuchen, welche Bausteinverbindungen hinsichtlich möglicher Konfigurationsänderungen zu berücksichtigen sind. Dabei dient die Bewegungsrichtung als eines der wichtigsten Entscheidungskriterien.

Es zeigt sich jedoch, daß gerade dieser Bewegungsverlauf nicht direkt anhand der vom System bereitgestellten Informationen festzustellen ist. Von vorn herein bekannt sind lediglich konkrete Bausteinpositionen, die, als Fixpunkte aufgefaßt, keine Rückschlüsse auf zeitliche Entwicklungen zulassen. Die Modellbewegung muß statt dessen unter Berücksichtigung zeitlicher Abfolgen von Positionsänderungen gesondert berechnet werden.

Diese Form der indirekten Informationsaufbereitung innerhalb eines zeitabhängigen Bezugsrahmens wirft ein weiteres Problem auf. Anhand der bekannten Positionsdaten ist zwar der bereits erfolgte Bewegungsverlauf eines Modells korrekt reproduzierbar, nicht aber die zu dem aktuellen Zeitpunkt tatsächlich stattfindende Bewegungsrichtung. Grund dafür ist die Tatsache, daß sich diese Richtung – es handelt sich hierbei um einen Vektor – nur anhand zweier Positionen feststellen läßt, von denen die eine zwangsläufig veraltet ist. Ein zu dem aktuellen Zeitpunkt gültiger Wert ist daher nicht eindeutig zu ermitteln.

Dieses Problem wird umgangen, indem eine Einschränkung der Bewegungsfreiheit vorgenommen wird, deren Auswirkungen sich in der Praxis als irrelevant herausstellen. Unterzieht man die Bewegungsbahn eines gegriffenen Modells einer genaueren Betrachtungen, läßt sich daran in der Regel ein kontinuierlicher Verlauf ohne abrupte Richtungsänderungen ablesen.

Diese Eigenschaft wird ausgenutzt, um eine Vorhersage der zu vermutenden Bewegungsrichtung zu treffen (siehe Abbildung 26).

Die Interpolation der Bewegungsrichtung r_0 erfolgt iterativ und basiert auf zwei bekannten Positionswerten P_1 und P_2 des gegriffenen Modells, wobei P_1 dem ältesten und P_2 dem neuesten Meßwert entspricht. Anhand dieser Werte läßt sich ein Vektor r_1 bestimmen, der die aktuelle Lageveränderung des Modells beschreibt und die Grundlage für die Bestimmung der angenommenen Bewegungsrichtung darstellt.

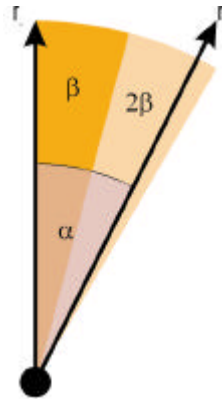


Abbildung 25: Geometrische Repräsentation der Bewegungsrichtung.

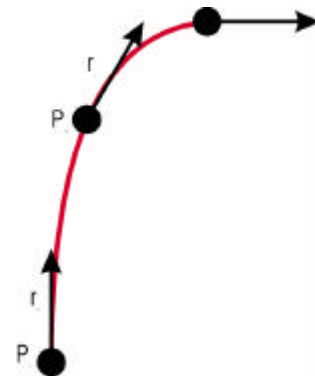


Abbildung 26: Verlauf einer Bewegung.

Zunächst wird überprüft, ob die zuletzt zurückgelegte Distanz, d.h. die Länge des Vektors r_1 größer als ein Grenzwert Δ ist. Trifft dies nicht zu, wird eine Ruhelage unterstellt, in der das Modell nicht bewegt wird. Gemessen wurden lediglich Positionsschwankungen, die zum Beispiel durch Handzittern oder Ungenauigkeiten des Tracking-Systems verursacht werden. Ist die Länge des Vektors hingegen größer als der vorgegebene Grenzwert, wird von einer bewußt ausgeführten Bewegung des Modells ausgegangen (siehe Abbildung 25).

In diesem Falle wird anhand des Winkels α , der sich zwischen dem Vektor r_1 und der bisher angenommenen Bewegungsrichtung r_0 aufspannt, entschieden, welche Bewegungsrichtung als derzeit gültig angenommen werden kann.

$$a = \vec{r}_1 \bullet \vec{r}_2 = a \cos(\left| \vec{r}_1 \right| * \left| \vec{r}_2 \right|), \quad \text{mit } \left| \vec{r}_1 \right| = \left| \vec{r}_2 \right| = 1$$

Ist dieser Winkel kleiner als ein fest vorgegebener Toleranzwinkel β , wird ein linearer Bewegungsverlauf unterstellt und r_0 bleibt als angenommene Richtung bestehen.

Trifft dies nicht zu, werden weitere Betrachtungen nötig. Da sich α nicht innerhalb des vorgesehenen Spielraums befindet, muß von einer Änderung der Bewegung ausgegangen werden. In dieser Situation werden zwei weitere Fallunterscheidungen vorgenommen, um abrupte Schwankungen zu unterdrücken. Erneut wird der Betrag des Winkels α überprüft. Stellt sich heraus, daß die Richtungsänderung auch größer als der zweifache Wert von β ist, wird als neuer Bewegungsvektor r_0' der Mittelwert von r_0 und r_1 festgelegt, um eine zu große

Abweichung von der bisher angenommenen Richtung auszugleichen. In dem verbleibenden Fall ($\alpha < 2\beta$) repräsentiert r_1 den neu anzunehmenden Richtungsvektor.

Im Zuge der Entwicklung wurden zwei zusätzliche Berechnungsverfahren der Richtungsbestimmung in Betracht gezogen:

$$\vec{r}_0' = \vec{r}_0 + \frac{1}{2}(\vec{r}_1 - \vec{r}_0)$$

Die direkteste Methode wäre eine Mittelwertberechnung, bei der sich der anzunehmende Vektor aufgrund des Durchschnitts einiger vorangegangener Positionswerte ergibt.

Bei dem zweiten Alternativansatz handelt es sich um die Methode der kleinsten Quadrate ([Sed1992], S.625), bei der ebenso auf eine Anzahl bekannter Werte zurückgegriffen wird. Zusätzlich wird hier eine Fehlerkorrektur hinsichtlich abweichender Positionswerte vorgenommen.

Beide Algorithmen haben den entscheidenden Nachteil gemeinsam, zu keinem Zeitpunkt die tatsächlich gemessene Bewegungsrichtung als Ergebnis zuzulassen. Indem grundsätzlich eine größere und damit veraltete Menge an Werten in die Berechnung einfließt, bleibt immer eine Differenz zwischen Eingangs- und Ausgangswert bestehen. Aufgrund dieser Trägheit konnte auf keines dieser Berechnungsverfahren zurückgegriffen werden.

4.2.2 Greifen und Zerlegen eines Modells

Wird innerhalb der Phase des Greifens ein Baustein aufgenommen, muß dieser zusammen mit weiteren, nicht abtrennbaren Elementen aus dem Ursprungsmodell herausgelöst werden.

Dieser Vorgang führt zu einem Zerfall des Ausgangsmodells in eine Anzahl von Bausteingruppen, die sich wiederum zu einzelnen Modellen zusammenfassen lassen (Siehe Kapitel 3.3.2.1). Das Verfahren zur Ermittlung dieser Ergebnismenge stellt in dieser Phase eine der wesentlichen funktionalen Aufgaben dar und wird im folgenden Abschnitt näher betrachtet.

Ein für den Zweck der Modelldemontage gut geeigneter Algorithmus stammt aus dem Bereich der Graphentheorie. Es handelt sich hierbei um die Methode der Tiefensuche, wobei sich die Implementierung der rekursiven Variante dieses Verfahrens bedient.

Zunächst kann festgestellt werden, daß es sich bei einem Modell um einen zusammenhängenden, ungerichteten Graphen handelt. Dabei entsprechen Bausteine den Knoten und bestehende Verbindungen den Kanten eines solchen.

Die Anwendung dieses Algorithmus setzt voraus, daß die, unter Berücksichtigung der Bewegungsrichtung bestehenden und ausgewählten Verbindungen der Modellbausteine bereits aufgehoben wurden. Das Modell ist in diesem Stadium nicht mehr als einzelner Graph aufzufassen, sondern repräsentiert einen Wald, d.h. eine Gruppe nicht zusammenhängender Graphen.

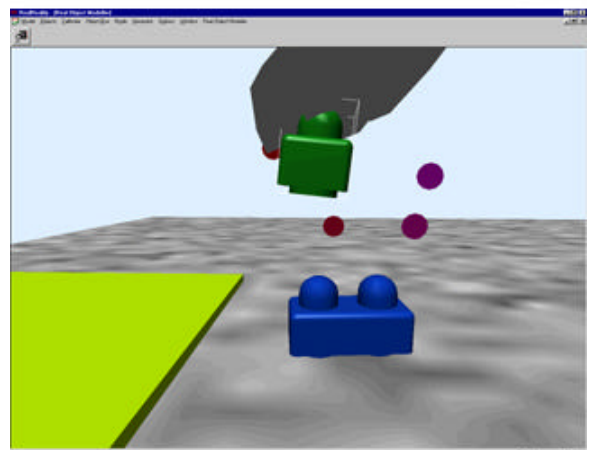
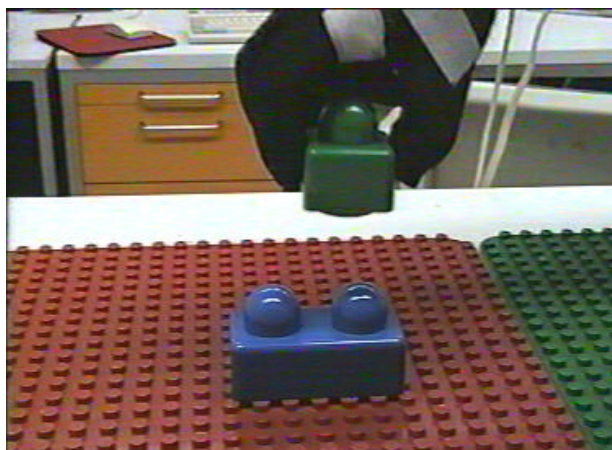
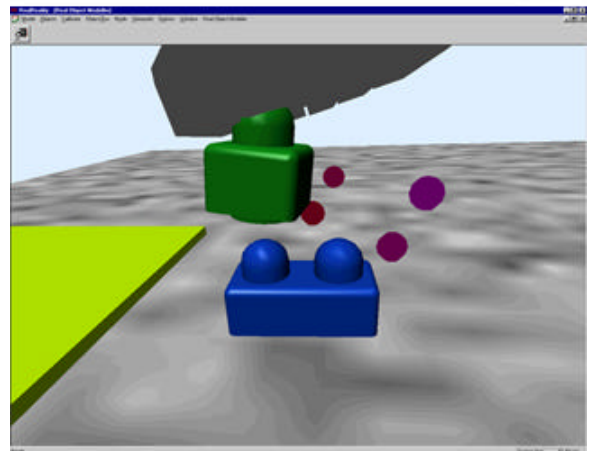
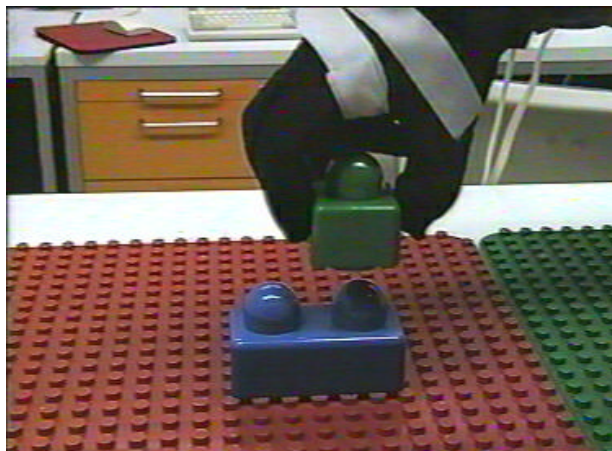
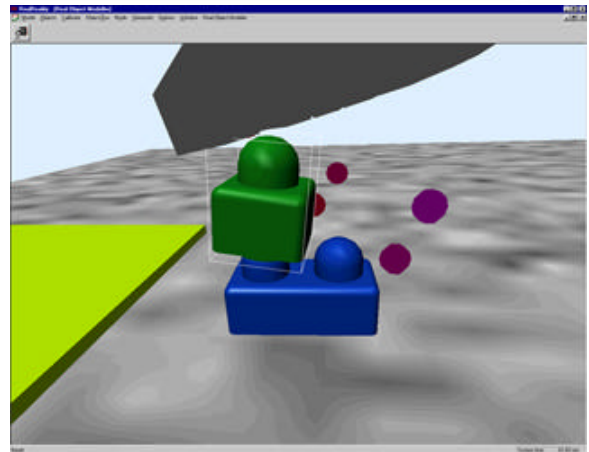
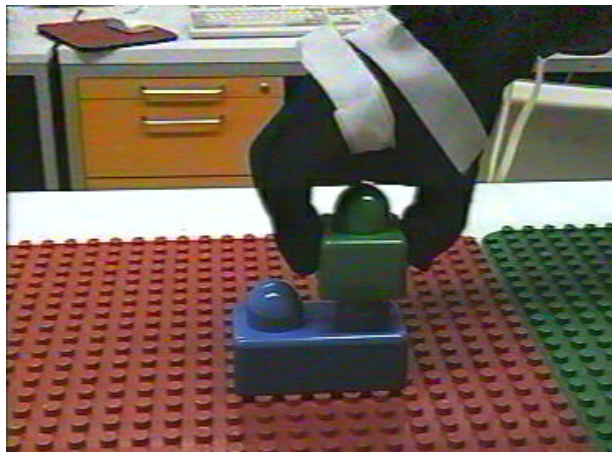
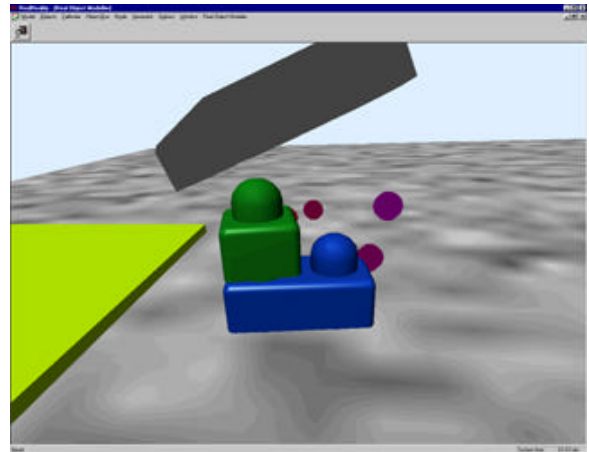
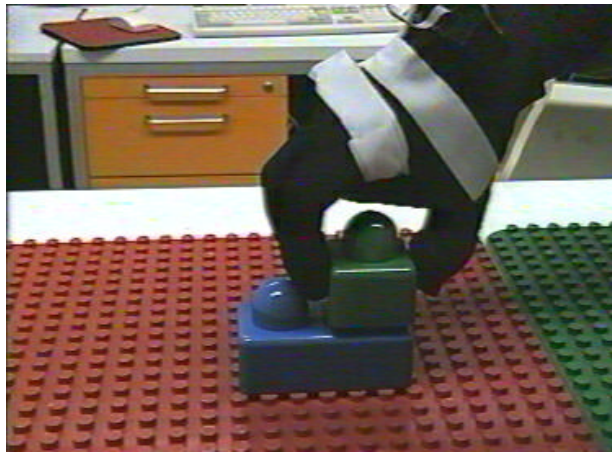


Abbildung 27: Greifen eines Bausteins und Heraustrennen aus dem Modell im Realen und Virtuellen.
 Bemerkung: Die zwei Ansichten verhalten sich spiegelverkehrt zueinander, da die reale Szene nicht vom Standpunkt des Modellieres aufgenommen wurde.

Die Tiefensuche erlaubt es nun, diese Graphen als Einheit innerhalb des Waldes zu identifizieren, indem alle existierenden Knoten (Bausteine) entlang ihrer Wege (Verbindungen) „besucht“ werden. Bei dieser Traversalion wird jeweils ein zusammenhängender Graph rekonstruiert, indem er als zusammenhängender Teil - Knoten für Knoten - wieder aus dem Wald herausgelöst wird. Um das gewünschte Ergebnis zu erhalten muß dieser Prozeß solange wiederholt werden, bis alle Knoten des Ursprungsgraphen mindestens einmal besucht wurden.

Da es sich bei dem Problem der Modelldemontage lediglich um einen kleinen Ausschnitt der gesamten Konstruktionsproblematik handelt, soll an dieser Stelle auf eine genauere Erläuterung der Tiefensuche verzichtet werden.

Eine Diskussion dieses und weiterer elementarer Graphenalgorithmen findet sich zum Beispiel in dem Buch „Algorithmen in C“ ([Sed1992], Kapitel 19).

4.2.3 Bewegen eines Modells

Während der Bewegungsphase im Rahmen der Anwendungsanalyse eine eher untergeordnete Funktion zukommt (Vgl. Abschnitt 3.3.2.2), ändert sich dieses Bild innerhalb der funktionalen Spezifikation grundlegend. Der Grund dafür liegt in der hohen Bedeutung, die der vollzogenen Bewegung des gegriffenen Modells innerhalb des Konstruktionsprozesses zukommt. Nahezu alle hier vorgestellten Algorithmen beziehen sich auf eine Vorauswahl verknüpfungsfähiger Elemente – und gerade diese läßt sich nur in Abhängigkeit der aktuellen Bewegungsrichtung treffen.

Eine Positionsänderung des gegriffenen Modells tritt ausschließlich während der Bewegungsphase auf. Aufgrund der Korrelation zwischen Position und Bewegungsrichtung kann auch die Vorauswahl der Elemente nur in diesem Abschnitt der Modellmodifikation ermittelt oder aktualisiert werden. Da sich die Position des gegriffenen Modells innerhalb dieser Phase in kurzen Zeitabschnitten verändert, sollte sich die daran gebundene Elementauswahl entsprechend zügig feststellen lassen. Dieser Forderung gilt es auch innerhalb einer komplexen Szene zu entsprechen, um einen dauerhaft flüssigen Simulationsablauf gewährleisten zu können. Aus diesem Grund werden bezüglich des Auswahlverfahrens eine Reihe von Mechanismen eingesetzt, die eine abgestufte Selektion potentieller Verbindungselemente ermöglicht.

Die dazu notwendigen Algorithmen sollen im weiteren Verlauf dieses Abschnittes vorgestellt werden.

4.2.3.1 Aktive Selektion

Generell können zwei unterschiedliche Formen der Elementauswahl unterschieden werden: eine aktive und eine passive Selektion verbindungsfähiger Bausteine. Die aktive Selektion bezeichnet dabei jene Elemente, die den Aufbau einer Verknüpfung initiieren, während passive Elemente unter Berücksichtigung ihres Zustands ein „Verbindungsangebot“ annehmen oder ablehnen können.

Die in diesem Abschnitt vorgestellte aktive Selektion bezieht sich generell auf die Bausteine des geprüften Modells, denn nur diese besitzen innerhalb der Szene die nötige Eigendynamik, um einen Verbindungsaufbau anstoßen zu können.

Darüber hinaus besteht eine direkte Abhängigkeit zwischen aktiver und passiver Selektion. Da von den aktiven Elementen das Erzeugen einer Verknüpfung bewirkt wird, stellen diese die Basismenge aller möglichen Verbindungen dar. Die passive Selektion wiederum erfolgt in Abhängigkeit dieser Elementbasis. Aufgrund dieser Zusammenhänge ist anzustreben, die Primärselektion über einen möglichst langen Zeitraum hinweg konstant zu halten, um eine vollständige Rekonstruktion der Gesamtauswahl zu vermeiden. Der Ansatz, die Bewegungsrichtung innerhalb einer vorgegebenen Toleranz unverändert bestehen zu lassen, leistet dazu einen wichtigen Beitrag.

Das Treffen der Auswahl potentieller Verbindungsoptionen unter Berücksichtigung der aktuellen Bewegungsrichtung (Direction of Movement / DoM) erfolgt anhand folgender Berechnung. Ein freier Verbindungspunkt des betrachteten Baustein ist dabei vorauszusetzen.

Zunächst wird der Winkel δ zwischen der Bewegungsrichtung (DoM) und der Verbindungsrichtung (DoC) mit Hilfe der normierten Vektoren berechnet:

$$d = \overrightarrow{DoC} \bullet \overrightarrow{DoM} = a \cos\left(\left|\overrightarrow{DoC}\right| * \left|\overrightarrow{DoM}\right|\right), \text{ mit } \left|\overrightarrow{DoC}\right| = \left|\overrightarrow{DoM}\right| = 1$$

Ist dieser Winkel kleiner als der vorgesehene Toleranzwinkel α des Anknüpfungspunktes, besteht für den Verbindungspunkt und damit auch für den Baustein die Möglichkeit eines Verknüpfungsaufbaus.¹ Anderenfalls ist eine Vorauswahl bezüglich des betrachteten Anknüpfungspunktes nicht möglich.

Um mit Hilfe dieses Verfahrens eine praktikable Gesamtauswahl zu erzielen, muß gewährleistet sein, daß der Toleranzwinkel aller Verbindungen innerhalb einer Szene nicht kleiner als der der Richtungstoleranz ist, um bei unveränderter Bewegung die aktive Auswahl beibehalten zu können.

¹ Der genaue strukturelle Aufbau der Verbindungspunkte ist in Kapitel 4.1.3 beschrieben

Nachdem die Elementauswahl des gegriffenen Modells abgeschlossen ist und diese nicht der leeren Menge entspricht, werden die verbleibenden passiven Modelle der Szene einer näheren Untersuchung unterzogen.

Hierbei handelt es sich um einen dreistufigen Prozeß, bei dem innerhalb jeder Stufe eine Verfeinerung der Zielauswahl vorgenommen wird. Am Ende dieses Vorgangs ist die derzeit aktuelle Auswahl der Zielelemente ermittelt. Die einzelnen Zwischenstufen dienen in darauffolgenden Iterationsschritten dazu, eine komplette Reorganisation der passiven Selektion zu umgehen, solange eine konstante Bewegungsrichtung vorliegt.

4.2.3.2 Vorselektion

Die Vorselektion erzeugt eine erste grobe Trennung der Bausteinelemente innerhalb der Szene. Die Auswahl wird anhand eines Sichtbarkeitstests vorgenommen, welcher die räumliche Lage eines Bausteins bezüglich des gegriffenen Modells feststellt. Unterschieden wird, ob sich ein Zielbaustein ausgehend von der Bewegungsrichtung vor oder hinter den Elementen der aktiven Selektion befindet. Dieses Verfahren ist an den Cyrus-Beck-Algorithmus zum „Clippen“ von Linien angelehnt (vgl. [Fol1990], S.117).

Zunächst wird die Szene mittels einer Ebene in zwei Halbräume unterteilt. Diese Ebene befindet sich orthogonal zur Bewegungsrichtung und liegt aus dieser Sicht direkt hinter dem letzten Baustein der aktiven Selektion. Alle Elemente, die sich vor dieser Ebene befinden, gelten als sichtbar, alle dahinter befindlichen sind unsichtbar und können deshalb von weiteren Betrachtungen ausgeschlossen werden. Ein Baustein ist genau dann sichtbar, wenn gilt:

$$\cos^{-1}(a) \geq 0, \text{ mit } \cos^{-1}(a) = \overrightarrow{N_e} \bullet \overrightarrow{DoM_N}$$

Wie auch die Selektion des gegriffenen Modells, steht die hier beschriebene Vorauswahl in direkter Abhängigkeit zur Bewegungsrichtung. Bei einer Änderung muß entsprechend auch hier eine Neuberechnung erfolgen.

Ändert sich die Bewegungsrichtung nicht, nimmt die Anzahl der Elemente innerhalb der Vorauswahl stetig ab. Durch die voranschreitende Bewegung des gegriffenen Modells werden vormals sichtbare Elemente in den unsichtbaren Halbraum der Szene verlagert und neue Elemente können nicht mehr hinzu kommen.

4.2.3.3 Sekundärselektion

Die Sekundärselektion stellt einen ersten Bezug zwischen der getroffene Vorauswahl sichtbarer Bausteine und der aktiven Selektion des gegriffenen Modells her. Es wird untersucht, ob sich innerhalb dieser Gruppen zueinander kompatible Elemente befinden. Dazu muß konkret festgestellt werden, ob Anknüpfungspunkte existieren, die in Anbetracht der gegebenen Bewegungsrichtung eine gegenseitige Verbindung aufbauen könnten.

Ziel ist es, unter Annahme einer konstanten Bewegungsrichtung die potentiell möglichen Zielelemente zu erfassen. Ist diese Auswahl einmal bekannt, wird - equivalent zur Vorauswahl - eine iterative Verfeinerung vorgenommen, so daß sich die Verarbeitungsgeschwindigkeit mit Abnahme der ausgewählten Element sukzessive verkürzt.

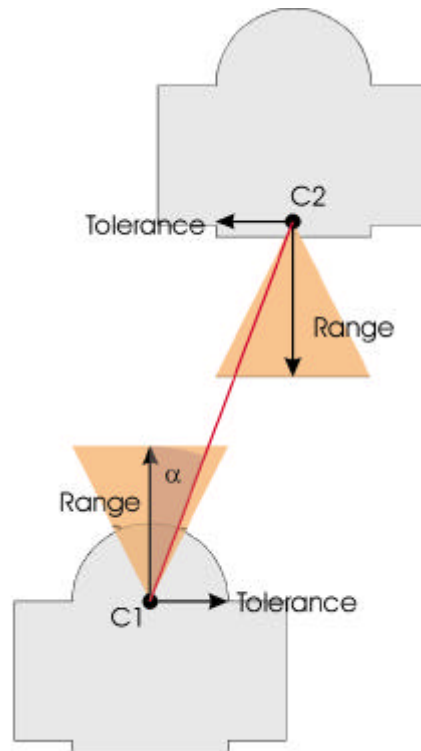


Abbildung 28: Geometrische Darstellung der Sekundärselektion.

Der für diesen Selektionsschritt vorgesehene Algorithmus ist zu dem der aktiven Elementauswahl nahezu identisch. Allerdings wird die dort betrachtete Bewegungsrichtung durch den Differenzvektor der beiden Verbindungspunkte ersetzt (siehe Abbildung 28). Zusätzlich müssen beide beteiligten Elemente hinsichtlich der Fähigkeit eines Verbindungsaufbaus überprüft werden.

Nachdem der Differenzvektor wie dargestellt bestimmt wurde, erfolgt die bereits bekannte Überprüfung des Toleranzwinkels anhand der Verbindungsrichtung und dem vereinheitlichten Differenzvektors.

4.2.3.4 Primärselektion

Die Primärselektion stellt die letzte Stufe der endgültigen Elementauswahl dar. Bausteine, die dieser Menge angehören, erlauben einen Verbindungsaufbau zu genau dem Zeitpunkt der Betrachtung. Konkret erfolgt innerhalb der Primärselektion eine Einschränkung der Sekundärselektion anhand des Auswahlkriteriums der Nähe. Dazu wird für jeden Baustein der Auswahl überprüft, ob sich sein potentieller Verbindungspartner innerhalb des aufgespannten Zylinders des betreffenden Anknüpfungspunktes befindet. Ist dies der Fall, handelt es sich hierbei

um eine verbindungsfähige Konstellation (siehe Abbildung 29). Da es sich bei dem Test auf Nähe lediglich um eine triviale Minimum-/Maximum-Überprüfung in lokalen Koordinaten der Verbindungspunkte handelt, kann hier auf eine explizite Notation verzichtet werden.

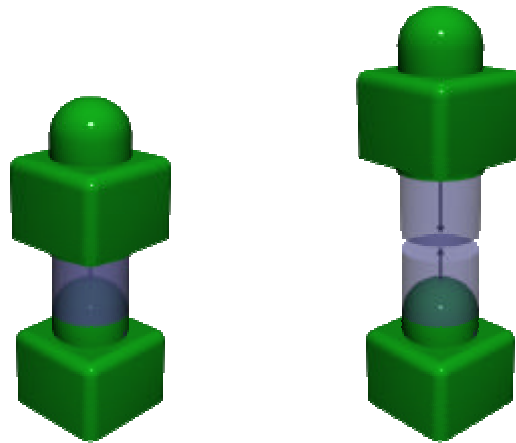


Abbildung 29: Das Kriterium der Nähe.
1. Fall: Bausteine sind sich nahe.
2. Fall: Bausteine sind nicht in unmittelbarer Nähe.

Da der Ermittlung der Primärselektion nicht nur die Bewegungsrichtung, sondern zusätzlich die genaue Position eines Elementes zugrunde liegt, läßt sich in diesem Fall kein iterativer Ausschluß realisieren. Es kann allerdings festgestellt werden, daß lediglich solche Elemente in die Primärselektion aufgenommen werden können, die zuvor Bestandteil der Sekundärselektion waren. Daher ist zu vermuten, daß es sich hierbei bereits um eine relativ geringe Anzahl von Bausteinen handelt, die zu der bereits bestehenden Auswahl hinzukommen können.

4.2.4 Loslassen eines Modells

Mit dem Loslassen des gegriffenen Modells setzt gleichzeitig die Phase des Verbindens ein, die sich im Wesentlichen durch zwei Fälle unterscheidet. Befindet sich eine Elementauswahl innerhalb der Primärselektion, muß eine Verbindung zwischen Modellen hergestellt werden, anderenfalls wurde das Modell unverändert abgesetzt.

Es wurde bereits angedeutet, daß eine Korrektur der Bausteinausrichtung sinnvoll erscheint, um Ungenauigkeiten im Rahmen der Positionserfassung abfangen zu können. Dafür vorgesehene Methoden werden an dieser Stelle vorgestellt.

4.2.4.1 Kombinieren von Modellen

Die Kombination eines gegriffenen mit einem Zielmodell richtet sich nach der bestehenden Primärselektion. Elemente dieser Modelle, die sich gegenseitig referenzieren, erlauben den Aufbau einer konkreten Verbindung und realisieren auf diese Weise die Vereinigung der Modelle. Kommt es an dieser Stelle zu Zweideutigkeiten, d.h. enthält die Primärselektion eines Modells mehrere Zielmodelle, ist anhand der geringsten Distanz zwischen den einzelnen Verbindungspunkten zu entscheiden, welcher der möglichen Partner zu favorisieren ist.

Nachdem die Eindeutigkeit einer Verbindung geklärt ist, gilt es eine korrekte Ausrichtung des beweglichen, d.h. gegriffenen Modells vorzunehmen. Auch in diesem Fall wird mit der Verknüpfung begonnen, die die geringste Distanz zwischen den einzelnen Verbindungspunkten aufweist.

In einem ersten Schritt sind zwei Korrekturen möglich. Zunächst wird die Verbindungsrichtungen des frei beweglichen Verbindungspunktes an die entsprechende Richtung des zweiten angeglichen. Danach wird ein Positionsausgleich vorgenommen, so daß die Verbindungspunkte letztendlich die gleiche Lage im Raum einnehmen. Als Ergebnis erhält man den Idealfall einer Verbindung, in der keinerlei Ungenauigkeiten mehr vorzufinden sind.

Trotz dieses Ausgleichs befinden sich die Elemente des gegriffenen Modells nicht zwangsläufig in einer korrekt ausgerichteten Lage. Erst durch den Ausgleich eines zweiten Verbindungspunktes (idealer Weise einer des gleichen Bausteins), erhält man die endgültige Lage der Bausteingruppe. Dabei ist zu beachten, daß die im ersten Bearbeitungsschritt vorgenommenen Anpassungen nicht wieder verworfen werden. Konkret bedeutet dies eine Einschränkung der Bewegungsfreiheit auf eine Rotation des Modells um die Achse des ersten Verknüpfungspunktes. Diese Ausgleichsmöglichkeit wird genutzt um das Modell so zu drehen, daß eine Übereinstimmung des zweiten Verknüpfungspaares gelingt. Der benötigte Rotationswinkel ergibt sich anhand des Vektorproduktes zwischen den Vektoren des bereits ausgerichteten Anknüpfungspunktes, dem Zielpunkt und dem auszurichtenden Verknüpfungspunkt.

4.2.4.2 Absetzen eines Modells

Dem Absetzen eines Modells kommt im Rahmen des Konstruktionsprozesses eine Sonderstellung zu. Einerseits ändert sich in diesem Falle die Modelltopologie einer Szene nicht, trotzdem hat eine Veränderung der Gesamtsituation stattgefunden, die es entsprechend zu berücksichtigen gilt.

Es ist zu bedenken, daß der Vorgang des Modellierens wesentlich von der präzisen Erkennung gegriffener Bausteine abhängt. Diese stützt sich wiederum auf eine korrekte und eindeutige Objektposition. Bauteile mit abweichender Position zur Realität, oder mit gegenseitigen Durchdringungen, können unter diesen Voraussetzungen nicht eindeutig erkannt werden und ziehen erhebliche Probleme im Umgang mit dem System nach sich.

Entsprechend zeigt sich, daß es sich bei den Maßnahmen der Kollisionserkennung und des Positionsausgleichs nicht primär um „kosmetische“ Korrekturen sondern vielmehr um notwendige Maßnahmen handelt, die einen korrekten Ablauf der Simulation gewährleisten sollen.

Der hier betrachtete Fall des Absetzens ist einer der Fälle, bei dem diese Ausgleichsmechanismen intensiv angewandt werden. In der Praxis hat sich gezeigt, daß gerade in dem Augenblick des Loslassens oftmals eine ungewollte Verschiebung des Modells eintritt, die in direktem Zusammenhang mit dem eingesetzten Griffserkennungsmechanismen stehen. Bereits frühzeitig zeichnete sich ab, daß diesem Problem mit entsprechenden Maßnahmen begegnet werden muß.

Der erste Lösungsansatz lehnt sich an die, im dem Bereich der graphischen Datenverarbeitung weit verbreitete Methode der Rasterung an. Im Rahmen dieses Verfahren wird der Aktionsraum nach einer festgelegten Schrittweite in ein entsprechendes Gitter unterteilt. Sämtliche darin befindlichen Objekte werden an diesem Raster ausgerichtet, so daß immer nur bestimmte Positionen im Raum von einem Objekt bezogen werden können. Die Implementierung dieses Verfahrens zeigte jedoch, daß diese Einschränkungen ein nicht zufriedenstellendes Ergebnis liefern. Ein Grund dafür ist zum Beispiel, daß mögliche Verbindungspunkte ebenfalls diesem Raster angeglichen werden müssen, um überhaupt ein passendes Partnerobjekt ausfindig machen zu können. Zusätzlich machten sich die Einschränkung der möglichen Objektpositionen im Zuge des Bewegungsprozesses unangenehm bemerkbar, da die Gegenstände ständig von der einen zur nächsten erlaubten Position sprangen. Den Ergebnissen entsprechend wurde dieser Ansatz nicht weiter verfolgt.

Statt dessen wurde die Beschaffenheit der Konstruktionsfläche ausgenutzt. Da es sich hierbei um eine fest vorgegebene, horizontale Ebene handelt, kann daraus geschlossen werden, daß sich abgesetzte Objekte - unabhängig von ihrer Höhenlage - auf dieser Ebene niederlassen und sich dort entsprechend horizontal ausrichten. Eine Rotation um die vertikale Raumachse bleibt dabei unberührt. Dieser Ansatz sollte sich in Kombination mit einer LEGO-Platte als Fixierhilfe als ausgesprochen praktikabel herausstellen.

Mit der hier endenden Beschreibung der Algorithmen findet eines der wichtigsten Kapitel dieser Arbeit seinen Abschluß. Ziel der hier vorgenommen Spezifikation war es einerseits, einen umfassen Überblick über die Gesamtstruktur des realisierten Prototypen zu geben. Andererseits wurden aber auch auf wichtige Details hingewiesen, um einen Eindruck der Komplexität dieser Aufgabenstellung zu vermitteln.

Kapitel 5

Geschichte einer Implementierung

Da im Rahmen der Anwendungsspezifikation bereits eine eingehende Betrachtung der wesentlichen Algorithmen und Datenstrukturen erfolgte, wird hier auf eine detaillierte Diskussion der Implementierung verzichtet. Zudem würde es den Rahmen dieser Arbeit bei weitem sprengen, tiefergreifende Betrachtungen eines Programmes vorzunehmen, dessen Umfang sich in der Größenordnung von 18.000 Zeilen Quellcode bewegt.¹

Statt dessen sollen einige hinsichtlich der Implementierung markante Aspekte aufgegriffen und aus meiner persönlichen Sicht dargestellt werden. Dazu werde ich den streng formalen Rahmen einer Diplomarbeit verlassen, um in einer etwas frechen und selbstkritischen Erzählung deutlich zu machen, daß bei der Entwicklung einer Software nicht allein formale Kriterien, sondern auch persönliche Eindrücke und (Fehl-) Einschätzungen eine wichtige Rolle spielen.

5.1 Entwicklungsrahmen

Bevor ich mit der eigentlichen Implementierung beginnen konnte, mußten zunächst einige grundlegende Rahmenbedingungen geklärt werden. Dazu gehörte zunächst, Entscheidungen über einzusetzende Software-Tools, die Frage der verwendeten Programmiersprache, verfügbare Programmbibliotheken usw. zu treffen. Aufgrund des Umfeldes, in dem ich den Prototypen realisieren wollte, ergaben sich bereits erste Tendenzen.

Da diese Arbeit in engem Kontakt zum Forschungszentrums Arbeit und Technik (artec) entstand und dort die nötigen technischen Voraussetzungen für eine Realisierung des geplanten Konzeptes gegeben waren, bot es sich an, das vorhandene Potential zu nutzen und sich entsprechend darauf einzustellen. Darüber hinaus lagen bereits fortgeschrittene Entwicklungen in hinsichtlich des Real Reality Konzeptes (vgl. [Bra1997b]) vor, so daß mir neben den rein technischen Mitteln auch die dort gesammelten Erfahrungen zugute kamen.

¹ Ebenso vermessen wäre es anzunehmen, es handle sich bei einer Implementierung diesen Umfangs um eine ausgereifte Applikation. Eine alte Programmierer-Weisheit besagt schließlich: Innerhalb einer prototypischen Entwicklung ist die Anzahl der Fehler immer echt größer als die Anzahl der Features.

Aus dieser Situation heraus entschied ich mich für den Einsatz der Microsoft Visual-C++ Entwicklungsumgebung. Die darin enthaltene Klassenbibliothek „Microsoft Foundation Classes“ kam ebenfalls zum Einsatz, um ein geeignetes Hilfsmittel zur Entwicklung unter Microsoft Windows zur Verfügung zu haben. Darüber hinaus griff ich auf die in dem artec genutzte 3D-Grafikbibliothek WorldToolKit (WTK) von Sense8 ([Sen@]) zurück, um die geplante Darstellung dreidimensionaler Szenen zu vereinfachen.

Zusätzlich taten sich unerwartete Perspektiven auf. Es stellte sich heraus, daß das im Rahmen des Forschungsprojektes RUGAMS¹ entwickelte System eine Anbindung des von mir zu entwickelnden Prototypen erlauben würde. Der in diesem Rahmen entstandene „Real Object Manager“ (ROM) war bereits zu diesem Zeitpunkt in der Lage, mit Hilfe des „Daten- und Gestenservers“ (DGS) erfaßte Veränderungen einer gegebenen, realen Situation zu verarbeiten und am Bildschirm entsprechend grafisch aufzubereiten. Im Zuge meiner Arbeit bedeutete die Möglichkeit, auf diese Applikation zurückgreifen zu können, eine erhebliche Reduzierung des Implementieraufwands. Was ich zu diesem Zeitpunkt jedoch unterschätzte, war die Randbemerkung, daß sich dieses Programm bereits seit zwei Jahren in der Entwicklung befand, d.h. bereits erhebliche Implementierarbeiten geleistet waren.

Geplant war zunächst eine offene Netzwerkarchitektur, die es unter Einsatz eines geeigneten Protokolls erlauben würde, relevante Informationen über die Dynamik einer Szene erhalten zu können. Diese Möglichkeit war für meine Zwecke nicht ausreichend, denn es war unschwer zu erkennen, daß auch die von mir geplante Anwendung Manipulationen an den Objekten einer Szene vornehmen würde. Deshalb mußte das eingesetzte Protokoll auch die Fähigkeit besitzen, Informationen in umgekehrter Richtung zu versenden. Gesagt, getan – unter der maßgeblichen Leitung von Martin Faust - damaliger Student und engagierter artec-Mitarbeiter, dem ich an dieser Stelle einen großen Dank für seine technische und mentale Unterstützung aussprechen möchte – wurde die Spezifikation des sogenannte DOXNative-Protokolls ([Rom1997]) erarbeitet, in der eben diese geforderten Eigenschaften der Szenenmanipulation vorgesehen wurden.

5.2 Technische Gerätschaft

Nachdem aus meiner Sicht die Frage der Systemstruktur geklärt war, galt es sich technischen Aspekten zuzuwenden. Meine ursprüngliche Vorstellung bestand darin, ein Tool zum Erzeugen von Modellen aus Fischertechnik-Bausteinen (siehe Kapitel 3.1) zu entwickeln, um einen, unter Aspekten der Konstruktion interessanten Ansatz zu bieten. Nach einigen hartnäckigen Experimenten - in dessen Kontext auch die Implementierung des Szenenrasters fällt

¹ Rechnergestützte Übergänge zwischen gegenständlichen und abstrakten Modellen produktionstechnischer Systeme.

- wurde dieses Vorhaben aufgegeben. Ausschlaggebender Faktor waren die technischen Gerätschaften, mit deren Hilfe die Erfassung der realen Situation erfolgte. Obwohl es sich bei diesem Equipment um neueste Entwicklungen handelte, war die für diesen Zweck erforderliche Präzision der Bewegungs- und Griffmerkennung nicht gewährleistet.

Einerseits konnte der eingesetzte Datenhandschuh von 5th Dimension Technologies nicht alle Freiheitsgrade der Hand abbilden, sondern beschränkte sich auf die Messung der Fingerkrümmung bei einer Auflösung von jeweils 256 Werten pro Finger. Andererseits erreichte das Tracking-System der Firma Polhemus bei einem Aktionsradius von ca. 70 cm Radius eine für Fischertechnik-Bausteine unzureichende Positionsgenauigkeit von 2,4 mm Abweichung im Mittel.

Aus dieser Erkenntnis heraus ergab sich für mich das Problem, ein neues, geeignetes Anschauungsobjekt ausfindig zu machen. Dabei kam mir unser kleiner Sohn hilfreich entgegen, denn er lenkte meine Aufmerksamkeit auf eines seiner Spielzeuge. LEGO-Primo – ein Baukastensystem für Kleinkinder – wies genau jene Eigenschaften auf, die ich für meinen Baukasten benötigte. Es ist voluminös, unkompliziert zusammenzufügen und bietet trotzdem ein umfangreiches Potential an Kombinationsmöglichkeiten. Warum also sollte man nicht ein Spielzeug für Kleinkinder in einem Bereich anwenden, dessen Entwicklung selbst noch in den Kinderschuhen steckt?

Nachdem auch dieses Problem gelöst war, ging es nun an die Aufgabe, ein geeignetes Konzept für die Implementierung auszuarbeiten und in die Tat umzusetzen.

5.3 Beobachtungen und Probleme

Die Ergebnisse meiner Überlegungen sind im Rahmen dieser Arbeit zur Genüge dargestellt worden. Was sich jedoch noch nicht gezeigt hat, sind die Umstände unter denen dieses Konzept umgesetzt wurde.

Nachdem feststand, was es zu implementieren galt und welche Struktur sich hinter dieser Problematik verbarg, begann ich mir konkrete Gedanken über den Verlauf der Implementierung zu machen. Zunächst erschien es mir wichtig, überhaupt die Struktur eines Baukastensystems festlegen zu können, denn ohne eine genaue Definition der Bausteine ließe sich auch keine Simulation einer Konstruktion durchführen.

Der nächste Schritt meiner Überlegungen war folgender: Im Rahmen einer Anwendung, die einen natürlichen Umgang mit dem Computer in den Mittelpunkt stellt, darf es nicht so sein, daß die dazu notwendigen Vorabinformationen in schriftlicher Form, d.h. innerhalb einer Textdatei, niederzulegen sind. Also entschied ich mich, unter Zuhilfenahme des World Tool Kit einen graphischen Editor zu implementieren, der die interaktive Definition eines virtuellen Baukastensystems ermöglicht.

Worüber ich mir dabei allerdings nicht klar wurde, war der damit verbundene Aufwand. Die bereits erwähnte Randbemerkung (2 Jahre Entwicklungszeit) wurde von mir geflissentlich übergangen und ich begann mit der Entwicklung, zumal auch noch nicht mit der Implementierung des Netzwerkprotokolls begonnen war. Ich fing also an, mich auf die Windows-Programmierung vorzubereiten - laut Fachliteratur ein Jahr Einarbeitungszeit (siehe [Kru1996]).

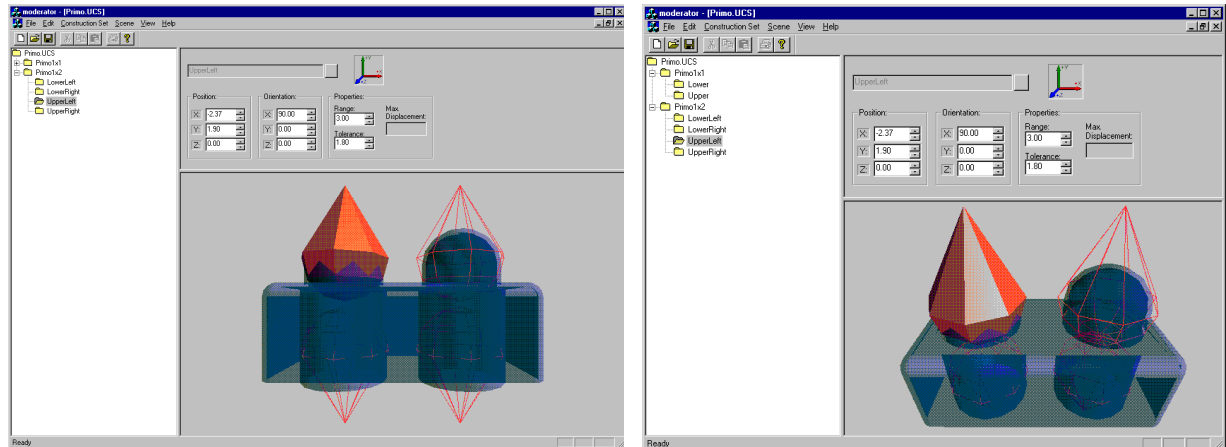


Abbildung 30: Darstellung eines Anknüpfungspunktes im grafischen Editor MODERATOR (Modell Generator).

Die Entwicklung machte Fortschritte, man konnte Bausteintypen definieren, Anknüpfungspunkte setzen und hatte anhand einer dreidimensionalen Ansicht gleichzeitig eine Kontrollmöglichkeit der gesetzten Parameter. Auch das Abspeichern und Einladen einer Baukasten-Datei funktionierte zuverlässig (siehe Abbildung 30). Allerdings stellte sich die Programmierung der graphischen Benutzungsschnittstelle insgesamt als ausgesprochen umfangreich dar, zumal es konzeptionellen Unterschiede zwischen der in C erstellten WTK-Bibliothek und der C++ - Anwendungsumgebung zu berücksichtigen und auszugleichen galt. In Bezug auf das Netzwerkprotokoll gab es ebenfalls Fortschritte zu verzeichnen, es konnte also mit der Implementierung des Konstruktionsprozesses begonnen werden. Inzwischen war ein Jahr verstrichen!

In Anbetracht der fortgeschrittenen Zeit entschloß ich mich, auf eine weitere Ausarbeitung des Modellbaukasten-Editors zu verzichten, obwohl dieser noch einige „offene Enden“ aufwies. In einem weiteren Schritt ging es nun darum, das Netzwerkprotokoll in den Prototypen zu integrieren. Keine aufwendige Sache – schließlich existierte bereits eine Bibliothek, auf der man aufsetzen konnte. Entsprechend folgten die Implementierung und das Testen einschließlich Fehlersuche. Zwei weitere Monate gingen vorüber und kein einziger Baustein war bis jetzt zusammengesetzt worden.

Nachdem auch die Kommunikation mit dem ROM realisiert war, sollten die bereits erdachten Algorithmen in eine konkrete Form gebracht werden. Also wurde weiter entwickelt, angefangen bei der Szene, über die Modelle und Bausteine, bis hin zur Verknüpfung. Dabei trat

ein Problem zutage, das ein weiteres Mal einen erheblichen zeitlichen Aufwand nach sich zog. Zu kaum einem Augenblick der eigentlichen Codierung bestand die Möglichkeit, das entstehende Programm und die darin enthaltenen Strukturen auf ihre Korrektheit bzw. Funktionstüchtigkeit zu überprüfen. Aufgrund der starken gegenseitigen Abhängigkeiten war es kaum möglich, einzelne Teile isoliert zu betrachten und zu testen. Erst nachdem die gesamte funktionale Basisstruktur fertiggestellt war, konnte mit der immer umfangreicher werdenden Fehlersuche begonnen werden. Obwohl ich die theoretische Grundlage der Implementierung wieder und wieder überprüft hatte, blieben Fehler natürlich nicht aus und es mußten immer wieder Änderungen an dem Quellcode vorgenommen werden. Erschwerend kam hinzu, daß der eigentliche Prozeß der Fehlersuche in Kombination mit dem ROM stattfinden mußte, so daß sich die vorhandenen Hilfsmittel, wie zum Beispiel der Debugger, nur begrenzt einsetzen ließen. Auch konnte nicht in allen Fällen von der Fehlerfreiheit des ROM ausgegangen werden, so daß es eine zusätzliche mögliche Fehlerquelle zu berücksichtigen galt.

Zu guter Letzt gelang es, nach einer Entwicklungszeit von über zwei Jahren, den Prototypen in einen funktionstüchtigen Zustand zu versetzen. Funktionstüchtig bedeutet in diesem Zusammenhang nicht etwa, das es sich hierbei um eine fehlerfreie Anwendung handelt. Gemeint ist vielmehr ein lauffähiger Prototyp, an dem die prinzipielle Realisierbarkeit und Tauglichkeit des von mir entwickelten Konzeptes demonstriert werden kann. Obwohl ich mir gewünscht hätte, noch eine etwas stabilere Version realisieren zu können, wurde an diesem Punkt der Entwicklungsprozeß eingestellt, um sich dem verbleibenden Teil dieser Diplomarbeit zuzuwenden.

5.4 Selbsteinschätzung

Möglicherweise wird sich die Mehrheit der Leser bei dieser Schilderung an den Kopf fassen. Ich möchte bezüglich dieser Beschreibung jedoch nicht zu dem Schluß kommen, daß die Umsetzung des in dieser Arbeit vorgestellten Konzeptes aufgrund der zum Teil selbstverschuldeten Probleme gescheitert sei. Eine solche Behauptung wäre schlichtweg falsch und ließe sich sicherlich anhand des Prototypen widerlegen.

Allerdings zeigt sich hier recht deutlich, daß die Realisierung eines solchen Systems aus Sicht der Implementierung als nicht trivial und vor allem sehr zeitaufwendig einzustufen ist. Dabei ist die Versuchung, ein bestehendes Problem soweit zu reduzieren, bis es sich als „einfach“ darstellt, recht hoch. Der tatsächliche Aufwand und die damit verbundenen technischen Probleme werden oftmals unterschätzt.

Dies sind zumindest meine Erfahrungen, die ich persönlich im Laufe der Implementierung gesammelt habe und die mich in Zukunft hoffentlich vor weiteren Fehleinschätzungen bewahren.

5.5 Perspektiven

Bezüglich der Funktionalität des Prototypen – von der Laufzeitstabilität einmal abgesehen – kann zunächst festgestellt werden, daß die angestrebten Ziele im Wesentlichen erreicht wurden. Daß es praktisch möglich ist, ein Modell aus Primo-Bausteinen zusammenzusetzen, zeigen unter anderem die Abbildungen dieser Arbeit. Es zeigten sich allerdings leichte Schwächen bezüglich der eindeutigen Erkennung der Bewegungsrichtung, wenn Bausteine sehr langsam zusammengesetzt wurden. Entsprechend scheiterte in diesen Fällen die Erkennung des Montagevorgangs. Hier gäbe es einen ersten Korrekturbedarf.

Das im Rahmen der Kommunikation mit dem ROM genutzte, verbindungsorientierte Netzwerkprotokoll stellte sich in der Praxis jedoch als zu langsam heraus. Die erreichten Übertragungsgeschwindigkeiten reichen für eine flüssige Simulation des Modellierprozesses nicht aus, so daß sich an dieser Stelle eine entsprechende Optimierung anbieten würde.

Darüber hinaus wäre es aus den bereits genannten Gründen, die ich nach wie vor vehement vertrete, wünschenswert und sinnvoll, auch den Baukasten-Editor in seiner Funktionalität zu vervollständigen und gegebenenfalls weiter auszubauen.

Eine weitere Einschränkung wurde hinsichtlich der abbildbaren Anknüpfungspunkte vorgenommen. Zur Zeit sind nur einfache Verbindungen zwischen zwei Bausteinen möglich. Zudem besitzen diese eine grundsätzlich starre Struktur, die keine Bewegung der Einzelteile (wie dies z.B. bei einer Welle erforderlich wäre) zulassen. An dieser Stelle gäbe es diverse Ansätze zu verfolgen, um zusätzliche Verbindungsprinzipien zu realisieren, um auf diesem Weg die Einsatzmöglichkeiten der Anwendung weiter auszubauen.

Innerhalb der Anforderungsdefinition wurde bereits auf funktionale Aspekte von Baukastensystemen hingewiesen. Auf diese konnte zwar im Rahmen der Arbeit nicht näher eingegangen werden, es ist aber meine Auffassung, daß sich in dieser Hinsicht eine Reihe interessanter Perspektiven eröffnen. Ein hier möglicher Ansatz wäre zum Beispiel der Einsatz von Constraints, wie ihn M. Gleicher in seiner Arbeit „A Differential Approach to Graphical Interaction“ vorstellt ([Gle1994]).

Eine weitere interessante Perspektive eröffnet sich, wenn man von einer direkten Abbildung der Bausteine innerhalb der Simulation absieht. Würde man die virtuelle Darstellung der Bausteine statt dessen durch konkrete Dinge des Alltags - wie zum Beispiel Möbelstücke - ersetzen, so ließe sich mit Hilfe des Baukasten ein Szenario konstruieren, dessen eigentliche Aussagekraft anhand des virtuellen Modells deutlich würde.

Beispiele für solche Anwendungen gäbe es viele. So wäre es z.B. denkbar, die funktional hoch optimierte Innenausstattung eines Wohnwagens oder Wohnmobils mit Hilfe eines

solchen Systems zu Gestalten¹. Auch die Planung von Küchen, Werkstätten usw. ließe sich mit Hilfe eines solchen Systems durchführen.

In einem weiteren Schritt könnte sogar die Konstruktion eines ganzen Hauses gelingen. Dazu müßte allerdings eine Art Geometrieanalyse realisiert werden, um Bausteingruppen nicht mehr als eine Anzahl von Einzelteilen aufzufassen, sondern als ein einziges Objekt. In einem solchen Fall könnte man eine Mauer beispielsweise aus einer Anzahl von Bausteinen zusammensetzen und das System würde daraus ein großflächiges Objekt - eben jene Mauer - generieren.

Mit diesem Ausblick soll die hier vorgestellte Arbeit abgeschlossen werden. Es hat sich gezeigt, daß das Konzept des synchronen Generierens von Modellen in realen und virtuellen Räumen einerseits eine große technische und konzeptionelle Herausforderung darstellt, andererseits aber auch Potentiale bezüglich des Umgangs mit Computern besitzt, welche sich in anderen Ansätzen nicht in dieser direkten Form wiederfinden lassen.

Mit dieser Arbeit wurde ein Versuch unternommen, sich diesem breiten Problemfeld anhand einer exemplarischen Betrachtung anzunähern. Aus persönlicher Sicht halte ich diesen Versuch trotz einiger Probleme für gelungen.

Da die hier vorgestellte Arbeit einen eher grundlegenden Ansatz zur Überprüfung der prinzipiellen Realisierbarkeit verfolgt, trat für mich die Frage nach einem konkreten Anwendungsfall ein wenig in den Hintergrund. Es wäre aus meiner Sicht jedoch erfreulich, wenn aufgrund der hier vorgestellten Methoden weitere anwendungsspezifische Entwicklungen angeregt werden.

¹ Im Rahmen einer Messedokumentation wurde tatsächlich ein Holzbaukasten vorgestellt, der speziell für den Zweck der individuellen Planung eines Wohnmobils konstruiert war.

Anhang

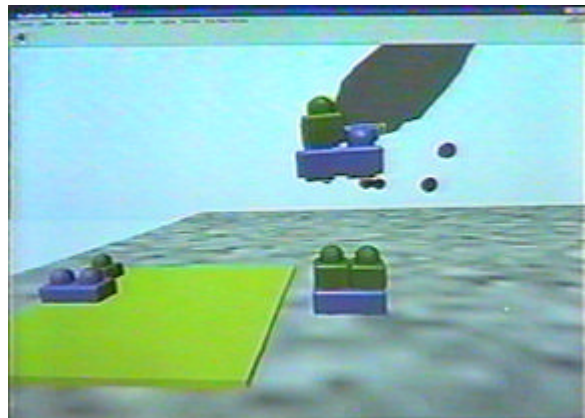
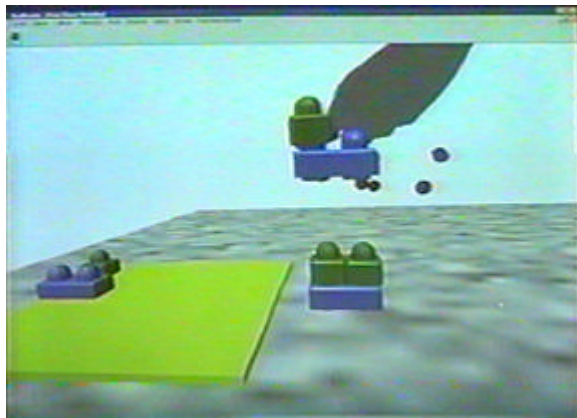
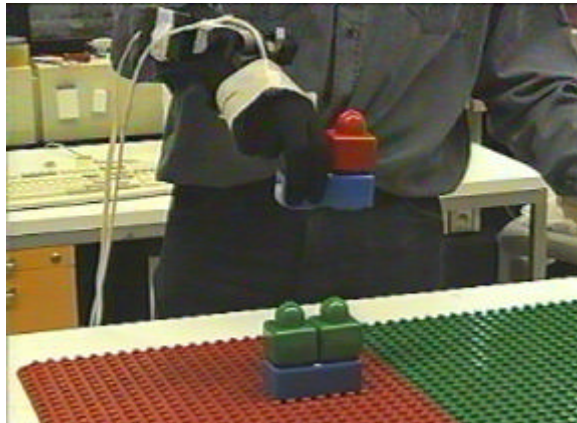


Abbildung 32: Bewegen eines zweiteiligen Modells.

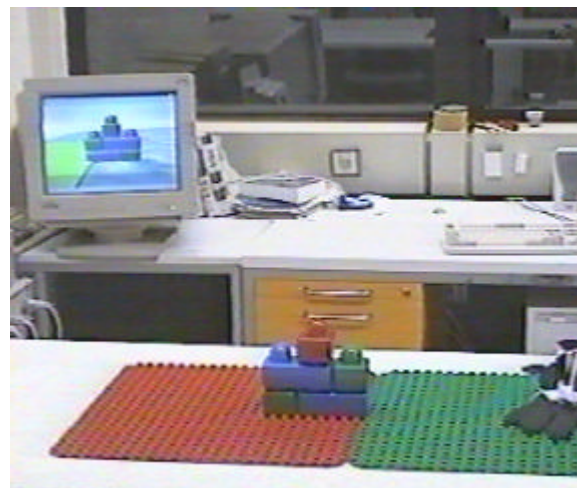
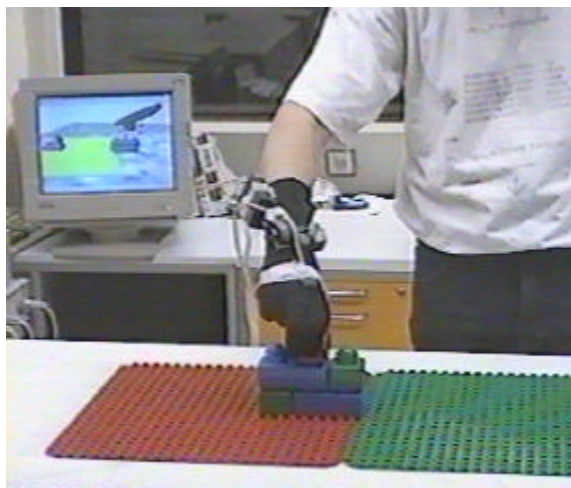


Abbildung 31: Zusammenbau eines wandartigen Modells.

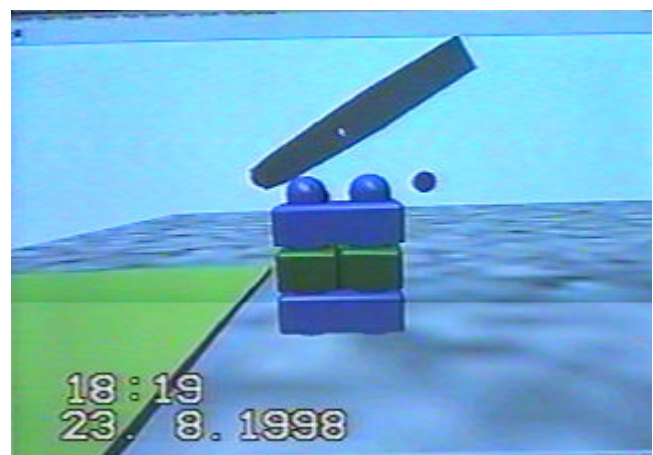
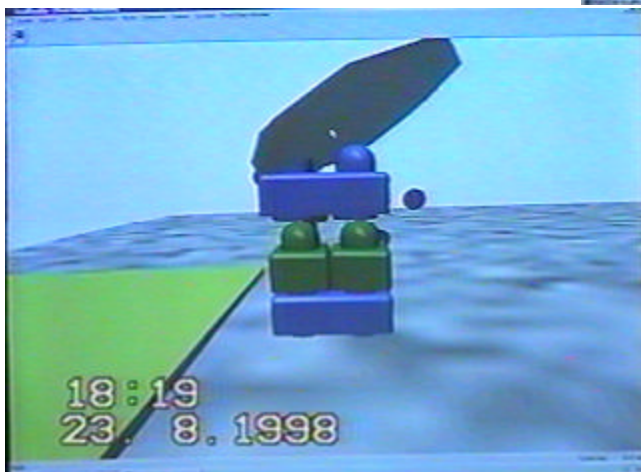
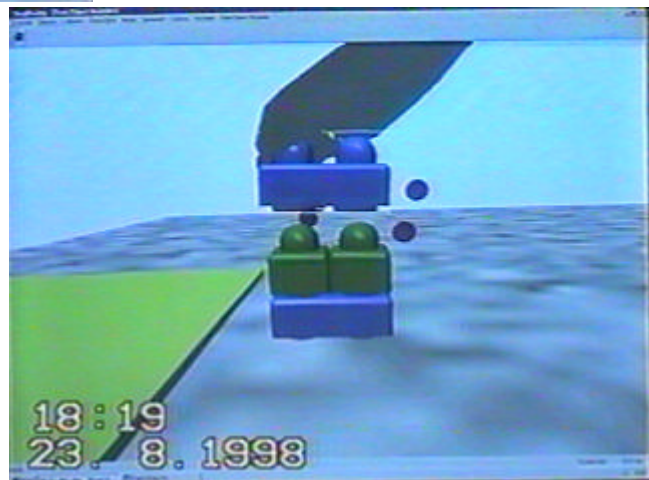
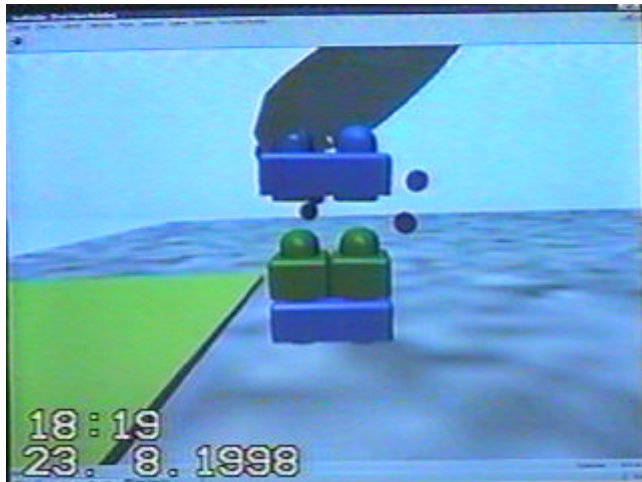


Abbildung 33: Zusammensetzen mit mehreren Verbindungspunkten.



Abbildung 34: Zusammenbau eines Modells

Literaturverzeichnis

- Ame1996 Ames, L.A., Nadeau D. R., Moreland J. L.
The VRML sourcebook.
John Wiley and Sons, New York 1996
- Bra1994 Brauer, V.
Feature-basierte Erkennung dynamischer Gesten mit einem Datenhandschuh, Diplomarbeit,
Universität Bremen, Bremen 1994
- Bra1997 Brauer V., Bruns F. W., Schäfer K.
Real Reality: Eine Synthese aus Simulation und Virtual Reality mit Anwendungsbeispielen
aus der Produktionstechnik (Beitrag zur Tagung der deutschen VR-Anwender in Köln, 10-12
Juni 1997), Universität Bremen, Bremen 1997
- Bra1997b Brauer, V., Faust, M.
Dokumentation zu den Systementwicklungen im Rahmen des DFG-Forschungsprojekts
RUGAMS, artec, Universität Bremen 1997
- Bri1990 Bricken, W.
Virtual Reality: Directions of Growth
Notes from the SIGGRAPH'90 Panel, 1990
<http://www.hitl.washington.edu/publications/r-90-1/>
- Bro1998 Broll, W.
Ein objektorientiertes Interaktionsmodell zur Unterstützung verteilter virtueller Umgebungen
Dissertation der Fakultät für Informatik der Eberhard-Karls-Universität zu Tübingen, GMD
Research Series, 1998, No. 11, GMD –Forschungszentrum Informationstechnik GmbH, Sankt
Augustin 1998
- Bru1993a Bruns, F. W.
Über die Rückgewinnung von Sinnlichkeit – Eine neue Form des Umgangs mit Rechnern,
Technische Rundschau Heft 29/30, Januar 1993
- Bru1993b Bruns, F. W., Heimbucher, A., Müller, D.
Ansätze einer erfahrungsorientierten Gestaltung von Rechnersystemen für die Produktion
artec-paper 21, Universität Bremen, März 1993
- Bru1996a Bruns, F. W., Brauer, V.
Bridging the Gap between Real and Virtual Modeling – A new Approach to Human-Computer
Interaction, artec-paper 59, Universität Bremen, Bremen 1996
- Bru1996b Bruns, F.W.
Modellierung und Programmierung technischer Anlagen durch gegenständliches Vormachen,
Forschungszentrum Arbeit und Technik (artec), Universität Bremen, Bremen, 1996
- Bru1996c Bruns, F.W.,
Grasping, Communicating, Understanding – Connecting Reality and Virtuality, artec-Paper
Nr. 44, artec, Universität Bremen, Bremen 1996
- Bru1998 Bruns, F.W.
Eröffnung des Workshops *Vom Bildschirm zum Handrad*
In: Bruns, F.W., Hornecker, E., Robben, B., Rügge, I.
Vom Bildschirm zum Handrad. Computer(be)nutzung nach der Desktop-Metapher, artec-
paper 59, Universität Bremen, Bremen 1997, S. 3-9

- Fai1993 Fairchild, K. M.
Information Management Using Virtual Reality Based Visualizations
In: Wexelblat A., Virtual Reality, Applications and Explorations
Academic Press Professional, Boston 1993.
- Fit1995 Fitzmaurice, G. W., Ishii, H., Buxton, W.
Bricks: Laying the Foundations for Graspable User Interfaces, Proceedings of CHI'95
Conference on Human Factors in Computing Systems, S. 442-449, ACM, New York 1995
- Fit1996 Fitzmaurice, G. W.
Graspable User Interface, PhD Thesis, Department of Computer Science, University of
Toronto, Toronto 1996
http://reality.sgi.com/gordo_tor/papers/gf/Thesis.gf.html/Thesis.gf.html
- Fit1997 Fitzmaurice, G., Buxton, W.
An empirical evaluation of graspable user interfaces: Towards specialized space-multiplexed
input
In: Proceedings of the 1997 ACM Conference on Human Factors in Computing Systems, CHI
'97, S. 43-50. ACM, New York 1997
- Fol1987 Foley, J. D.
Neuartige Schnittstellen zwischen Mensch und Computer
In: Spektrum der Wissenschaft, 1997
- Fol1990 Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F.
Computer Graphics – Principles and Practice, 2nd Ed., Addison-Wesley Publishing Company:
Massachusetts 1990
- Gin1995 Gindling J., A. Ioannidou J., Loh O., Lokkebo A.
LEGOsheets: A Rule-Based Programming, Simulation and Manipulation Environment for the
LEGO Programmable Brick
In: Proceeding of Visual Languages, S. 172-179, IEEE Computer Society Press, Darmstadt
1995
<http://www.cs.colorado.edu/~l3d/systems/legosheets/vl95paper/>
- Gle1994 Gleicher, M. L.
A Differential Approach to Graphical Interaction (CMU-CS-94-217), School of Computer
Science, Carnegie Mellon University, Pittsburgh 1994
- Hut1986 Hutchins, E. L., Holland, J. D., Norman, D.A.
Direct Manipulation Interfaces
In: Norman, D. A., Draper, S. W., User Centered System Design, S. 87-124, Lawrence
Erlbaum Associates, Hillsdale, New Jersey 1986
- Hys1997 Bruns, F.W., Müller, D., Steenbock, J.
Hypermediagestützte Simulationssysteme für berufliche Schulen. 3. Zwischenbericht zum
Modellversuch. Senator für Bildung und Wissenschaft, Bremen 1995
- Kru1996 Kruglinski, D. J.
Inside Visual C++ - Version 4.0, Microsoft Press, Redmond, Washington 1996
- Lau1986 Laurel, B. K.
Interface as Mimesis
In: Norman, D. A., Draper, S. W.
User Centered System Design, Lawrence Erlbaum Associates, Hillsdale, New Jersey 1982

- Laur1997 Lauria, R.
Virtual Reality: An Empirical-Metaphysical Testbed
School of Journalism and Mass Communication, University of North Carolina at Chapel Hill,
North Carolina 1997
<http://jcmc.msc.huji.ac.il/vol3/issue2/lauria.html>
- Pso1995 Psotka, J.
Immersive Tutoring Systems: Virtual Reality and Training, U.S. Army Research Institute,
Alexandria 1995
<http://alex-immersion.army.mil/its.html>
- Rau1997a Rauterberg, M., Bichsel, M., Meier, M. & Fjeld, M.
A gesture based interaction technique of a planning tool for construction and design.
In: Proceedings 6th IEEE International Workshop on Robot and Human Communication--RO-
MAN'97, (IEEE Catalog Number: 97TH8299), S. 212-217), Institute of Electrical and
Electronics Engineers, Piscataway 1997.
[Http://www.ifap.bepi.ethz.ch/~rauter/publications/ROMAN971paper.pdf](http://www.ifap.bepi.ethz.ch/~rauter/publications/ROMAN971paper.pdf)
- Rau1997b Rauterberg, M.
Natural User Interface (NUI): a case study of a video based interaction technique for a
computer game.
In: Smith, M.J., Salvendy G., Koubek R.J. (Eds), Design of Computing Systems: Social and
Ergonomic Considerations – HCI' 97 (Advances in Human Factors/Ergonomics), Vol. 21B,
S. 253-256, Elsevier, Amsterdam 1997.
[Http://www.ifap.bepi.ethz.ch/~rauter/publications/HCI97paper2.pdf](http://www.ifap.bepi.ethz.ch/~rauter/publications/HCI97paper2.pdf)
- Rau1998 Rauterberg M., Bichsel M., Fjeld M., Leonhardt U., Krueger H., Meier M.
BUILD-IT: a planning tool for constructioun and design.
CHI'98 conference companion.
[Http://www.ifap.bepi.ethz.ch/~rauter/publications/CHI98video.pdf](http://www.ifap.bepi.ethz.ch/~rauter/publications/CHI98video.pdf)
- Res1994 Resnick M.
Behavior Construction Kits
In: Communications of the ACM, Vol.37, No.7, S. 65-71, July 1994
- Rob1997 Robben, B., Rügge, I.
Real Reality: . Computer(be)nutzung nach der Desktop-Metapher
In: Bruns, F.W., Hornecker, E., Robben, B., Rügge, I.
Vom Bildschirm zum Handrad. Computer(be)nutzung nach der Desktop-Metapher, artec-
paper 59, Universität Bremen, Bremen 1997, S.159-173
[Http://www.artec.uni-bremen.de/field1/Workshop97/Beiträge/Robben.html](http://www.artec.uni-bremen.de/field1/Workshop97/Beiträge/Robben.html)
- Rom1997 Faust, M.:
Real Object Modeler – Netzwerkprotokolle, artec, Universität Bremen, Bremen 1997
- Schäf1997 Schäfer, K., Brauer, V., Bruns, W.
A new Approach to Human-Computer Interaction – Synchronous Modeling in Real and
Virtual Spaces, Proceedings of the DIS (Designing Interactive Systems), Amsterdam, NL,
1997
[Http://www.artec.uni-bremen.de/field1/rugams/papers/DIS97Paper/DISPaper.htm](http://www.artec.uni-bremen.de/field1/rugams/papers/DIS97Paper/DISPaper.htm)
- Sed1992 Sedgewick, R.
Algorithmen in C, Addison-Wesley GmbH, Bonn; München; Paris [u.a.] 1992
- Shn1992 Shneiderman, B.
Designing the User Interface – Strategies for Effective Human-Computer Interaction, 2nd Ed.,
Addison-Wesley Publishing Company, Massachusetts 1992

- Sno1994 Snowden, D. N., West, A. J.
The AVIARY VR-system: A Prototype Implementation, 6th ERCIM workshop, Stockholm 1994
<http://www.cs.nott.ac.uk/~dns/vr/aviary/>
- Suz1995 H. Suzuki, Kato H.
Interaction-Level Support for Collaborative Learning: AlgoBlock -- An Open Programming Language
Proceedings of the Computer Supported Collaborative Learning (CSCL) Conference, University of Indiana, 1995
<http://www-cscl95.indiana.edu/cscl95/suzuki.html>
- Thal1991 Thalmann, N. M., Thalmann, D.:
New Trends in Animation and Visualization, John Wiley & sons, Chichester, NY, Brisan, Toronto, Singepore 1991
- Tur1998 Turner R., Song L., Gobbetti E.
Metis: An Object-Oriented Toolkit for Constructing Virtual Reality Applications, Submitted for publication in Computer Graphics Forum, University of Maryland Baltimore County, Baltimore 1998
<http://www.csee.umbc.edu/~turner/metis.html>
- Wel1993a Wellner P., Mackay W., Gold R.
Computer-Augmented Environments – Back to the Real World
In: Communications of the ACM, 36/7, S. 24-26, 1993
- Wel1993b Wellner P.
Interacting with Paper on the DigitalDesk.
In: Communications of the ACM, 36/7, S. 87-96, 1993
- Wex1993 Wexelblat, A.
The Reality of Cooperation: VR and CSCW
In: Wexelblat A., Virtual Reality, Applications and Explorations
Academic Press Professional, Boston 1993.

Internet-Adressen

Cal@ Home Page von Caligari
<http://www.caligari.com>

EUG@ Home Page vom DFG-Projekt: Erfahrungsorientierte Übergänge zwischen gegenständlichen und abstrakten Modellen für die berufliche Qualifikation (EUGABE)
<http://www.artec.uni-bremen.de/field1/eugabe/index.html>

Fit@ Home Page von George Fitzmaurice:
<http://www.dgp.toronto.edu/people/GeorgeFitzmaurice/>

HYS@ Home Page vom DFG-Projekt Hysim
<http://www.artec.uni-bremen.de/field1/Hysim/index.html>

Ktx@ Home Page von Kinetix
<http://www.ktx.com>

Rau@ Home Page von Matthias Rauterberg
<http://www.ifap.bepr.ethz.ch/~rauter/>

RUG@ Home Page vom DFG-Projekt: Rechnergestützte Übergänge zwischen gegenständlichen und abstrakten Modellen produktionstechnischer Systeme
<http://www.artec.uni-bremen.de/field1/rugams/index.html>

Sen@ Home Page von Sense8
<http://www.sense8.com>

Vrc@ <http://www.vrml.org>