Abschlußbericht des DFG-Projekts

„ Datenbankgestützte Koordinierung und Integration von Planungswerkzeugen im Baubereich (ArchE)“

Kennwort: „ Rechnerintegrierte Gebäudeplanung“
Lo 296 / 11-1, 11-2

Dezember 1997

1 Zielsetzung und Arbeitshypothesen

1.1 Einführung in die Projektziele

Den Produkterstellungsprozeß kennzeichnet heute die Forderung nach hoher Produktivität, kurzen Durchlaufzeiten und garantierter Qualität. Um dieses Ziel zu erreichen, müssen die einzelnen Schritte der Prozeßkette unter einer möglichst engen Kooperation aller beteiligten Experten reibungslos ineinandergreifen. In diesem Prozeß nimmt der Baubereich eine Sonderstellung ein. Es handelt sich hier nicht, wie etwa im Automobilbau, um Serienproduktionen, sondern um Einzelproduktionen in einem ständig neuen Entwurfs- und Bauumfeld. In diesem auch heute noch vorwiegend handwerklich geprägten Umfeld werden die Entwurfs- und Bauleistungen der beteiligten Experten (Architekt, Statiker, Klimaingenieur, Elektroingenieur usw.) weitgehend sequentiell abgearbeitet. Die Bestrebungen, den Entwurfsprozeß in Form eines kooperativen Gebäudeentwurfes zu integrieren und so eine möglichst frühzeitige parallele Beteiligung aller Experten zu erreichen, steckten zu Beginn der Projektlaufzeit noch weit in den Anfängen.

Das Vorhaben hatte nun gerade die rechnergestützte Integration unterschiedlicher Fachbereiche des Gebäudeentwurfs in eine gemeinsame Entwurfsplattform zum Ziel. Wir wollten uns insbesondere auf die ersten Schritte im Entwurfs- und Bauprozeß konzentrieren, da gerade hier ein beträchtlicher Teil der in den nachfolgenden Schritten benötigten Daten anfällt. Dabei lag der Fokus auf dem architektonischen Entwurf, da er eine Schlüsselposition innerhalb des Entwurfsprozesses einnimmt und besonders in den ersten Phasen, infolge der schwach strukturierten und einer starken Dynamik unterliegenden Daten, die härtesten Anforderungen an eine Computerunterstützung stellt. Würde es gelingen, ihn geeignet zu formalisieren, so wäre damit die Grundlage für die Formalisierung späterer Fachplaner gegeben, so daß diese in eine gemeinsame Entwurfsplattform integriert werden können. Dazu mußte ein einheitliches Modellierungs- und Verarbeitungskonzept die Durchgängigkeit von den ersten Phasen hin zu den späteren stärker strukturierten Entwurfs- und Ausführungsphasen gewährleisten.

Auch heute noch orientieren sich die Ansätze für eine Rechnerunterstützung überwiegend an der sequentiellen Abfolge der Entwurfs- und Bauleistungen. Es gibt hochentwickelte, spezialisierte Expertentools, die jedoch zum größten Teil Insellösungen darstellen. Versuche, die einzelnen Fachaspekte zu integrieren, zielen auf eine Vereinheitlichung der Modellbildung aller am Entwurf Beteiligten sowie auf die Definition von Schnittstellen und Protokollen. Da sich die verwendeten Datenmodelle in der Regel an den Objekten eines vorformulierten Produktes orientieren, kann man die Modellierung produktorientiert nennen. Die Erfahrungen zeigen nun, daß sich die beschriebene Integrationsmethodik lediglich bei einer sequentiellen Abfolge der Entwurfsschritte bewährt, da sich dann das Problem auf den Austausch von Daten beschränkt. Es ergeben sich jedoch Schwächen, wenn der Entwurf kooperativ ablaufen soll und gerade, wie in den ersten Phasen, mit vielen Unwägbarkeiten, spontanen Sprüngen und Rücknahmen von Entscheidungen behaftet ist.

Unseren Arbeiten legten wir daher einen neuartigen Ansatz zugrunde. Ausgangselemente für unsere Datenmodellierung sind nicht die Elemente des fertigen Produktes allein, sondern die in den ersten Entwurfsphasen typischen unscharfen, skizzenhaften Informationen, informellen Vereinbarungen usw., die die ersten Grundlagen für einen kooperativen Gebäudeentwurf darstellen. Im Laufe des Entwurfes werden diese unscharfen Informationen dann sukzessive über Entwurfsentscheidungen, zugeschaltete Constraints und Partialmodelle sowie die allgemeinen Meilensteine des Gebäudeentwurfs strukturiert, eingeschränkt und in Produktmodelle der bisherigen Ansätze überführt.

Der Reiz des Vorhabens lag somit in seiner interdisziplinären Anlage. Von seiten der Architekten wurden die Entwurfsprozesse eingebracht, von seiten der Informatiker das Denken in formalen Modellen, Prozessen und die Datenintegration.

1.2 Arbeitshypothesen

Wenn den geistig-kreativen Tätigkeiten des Architekten systematisch-formale Vorgehensweisen des Informatikers zugeordnet werden sollen, bedarf es eines gewissen Regelungsrahmens, in dem beide Seiten zu einem gegenseitigen Verständnis kommen. Dieser hat sich im Laufe des Vorhabens zu den folgenden vier Hypothesen über den architektonischen Entwurfsvorgang verdichtet. Sie stellen für die Architekten eine neuartige Entwurfsvorgehensweise dar, und sie dienten den Informatikern zugleich als Prämissen für ihre Arbeiten im Projekt. Auf der prototypischen Realisierung des Entwurfssystems wurde eine Validierung dieser Hypothesen durchgeführt, die für den betrachteten Einsatzbereich positiv ausfiel. Die unter den Prämissen entwickelten Konzepte behalten zudem aber auch unabhängig vom Validierungsergebnis ihre Relevanz, weil sie auch unter anderen Voraussetzungen interessante und einsetzbare Ergebnisse zum Fortschritt der Informatik beitragen.

Hypothese 1:

Der architektonische Entwurfsvorgang - gleichgültig wo er sich zeitlich, räumlich und im Detail befindet - ist bereichsorientiert innerhalb eines mehrdimensionalen Entwurfsraumes. Die Bereiche werden durch die Dimensionen wie geometrischer Ort, Zeit, Maßstab, Entwurfsaspekt etc. aufgespannt und erlauben alle Entwurfsentscheidungen einheitlich und eindeutig, über alle Phasen und Sichten des Entwurfes hinweg zu positionieren. Diese bereichsorientierte Modellierung bildet eine tragfähige Integrationsbasis für den kooperativen Gebäudeentwurfsprozeß.

Insbesondere für die Elemente der frühen Entwurfsphasen ist kennzeichnend, daß sie eher an räumlichen Bereichen und Vorgängen als an physikalischen Objekten festgemacht werden können. Eine Modellierung muß daher zunächst bereichs- und vorgangsorientiert sein und sich dann allmählich im Laufe des Entwurfes in ein produktorientiertes Modell umwandeln lassen. Sie muß den Übergang von den schwach strukturierten Daten zu Anfang des Entwurfes hin zu den stark strukturierten Daten zu seinem Ende schaffen. Grundlage unseres Ansatzes ist daher ein bereichsorientiertes Kernmodell, das u.a. durch die schrittweise Einbindung von produktorientierten Partialmodellen stärker strukturiert wird. Das IFIB verfügt über die Konzeption einer bereichs- und vorgangsorientierten Modellierung mittels eines mehrdimensionalen Modells. Ob es in der Lage ist, die formulierten Anforderungen zu erfüllen und eine Integrationsplattform für alle Fachplaner während des gesamten kooperativen Entwurfsprozesses sein kann, war im Laufe des Projektes zu überprüfen.

Hypothese 2:

Entwurfsentscheidungen werden als Einschränkungen des Entwurfsraums in Richtung auf das Zielprodukt aufgefaßt. Sie werden als sogenannte Constraints formalisiert und bereichsbezogen formuliert.

Datenstrukturen allein reichen nicht aus, um alles Wissen, alle Objekte, Vorgänge und Abhängigkeiten während des Entwurfsprozesses, insbesondere auch in den ersten Phasen, vollständig abzubilden. Für die Repräsentation von Entwurfswissen der beteiligten Experten wird daher, ergänzend zu den Partialmodellen gemäß Hypothese 1, eine Formalisierung mittels Constraints benötigt, die in den Entwurfsprozeß eingebunden und über geeignete, ausgefeilte Konsistenzsicherungsmechanismen überwacht werden müssen. Constraints werden dabei als Entwurfsentscheidungen betrachtet, die den Entwurfsraum schrittweise in Richtung auf das Zielprodukt hin einschränken.

Hypothese 3:

Der Übergang von schwach strukturierten zu stark strukturierten Daten erfolgt mit zunehmendem Entwurfsfortschritt durch die Integration von Partialmodellen.

Mit zunehmendem Entwurfsfortschritt liegen die Entwurfsdaten in immer stärker strukturierter Form vor, was sich durch die Verwendung von Partialmodellen mit einer

ausgeprägten Stukturierung im Entwurfsprozeß wiederspiegelt. Die schrittweise Integration dieser Partialmodelle in die Entwurfsdatenbasis ist daher von besonderer Bedeutung für die integrierte Datenhaltung zur Unterstützung des gesamten Entwurfsverlaufs.

Hypothese 4:

Der architektonische Entwurfsprozeß ist hochgradig iterativ. Diese Dynamik kann durch einen objektbezogenen History-Mechanismus geeignet unterstützt werden.

Charakteristisch für den Entwurfsprozeß ist das häufige Zurücksetzen und Revidieren schon getroffener Entwurfsentscheidungen in beliebigen Ausschnittsbereichen des mehrdimensionalen Entwurfsraumes. Diese Dynamik kann daher als eine inhärente Eigenschaft aller Entwurfsobjekte, sowohl der Daten als auch der Constraints, angesehen werden. Dieser Anforderung muß durch ein durchgängiges Konzept Rechnung getragen werden, indem jedem einzelnen Entwurfsobjekt eine Historie zugeordnet wird, die die Rücksetzbarkeit auf Objektebene ermöglicht.

1.3 Systemarchitektur

Die oben aufgestellten Arbeitshypothesen führten zu der in Abbildung 1 dargestellten Systemarchitektur.

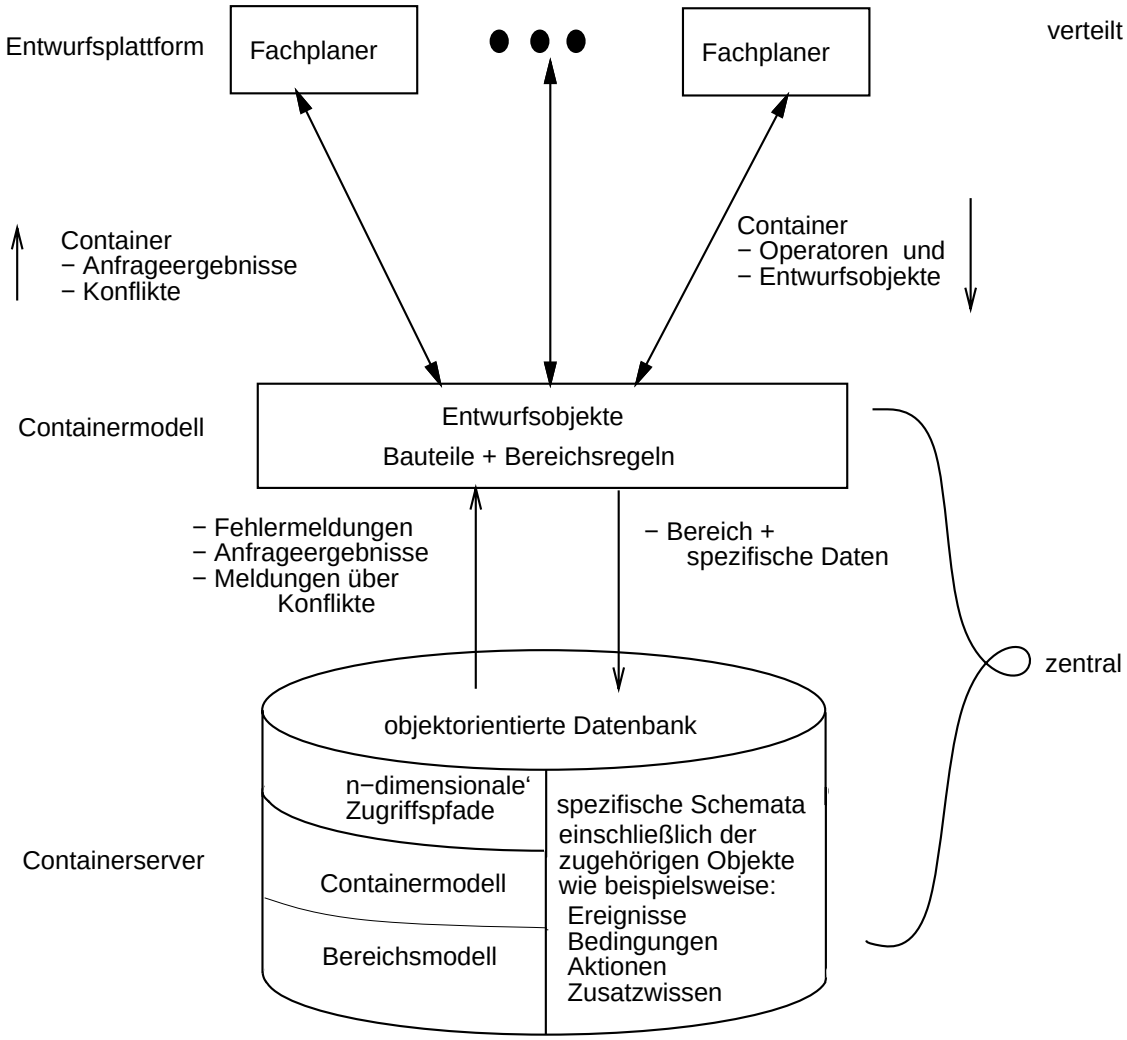


Abb. 1.1: *Prinzipielle Systemarchitektur*

Die einheitliche Betrachtungsweise aller Entwurfsentscheidungen als Bereiche innerhalb eines mehrdimensionalen Entwurfsraumes (Hypothese 1) führte zum Begriff des Containers (zur genauen Definition siehe Kapitel 2).

Das Erzeugen von Containern, ihre Manipulation und Kontrolle, sowie die Navigation der Planer durch den Entwurfsraum erfolgt über die Entwurfsplattform. Die dafür benötigten Werkzeuge werden in Form einer multimedialen Arbeitsumgebung allen Planern zur Verfügung gestellt.

Die Umsetzung des mehrdimensionalen Entwurfsraumes auf ein Datenmodell erfolgt im Containermodell. Die Datenhaltung der Container des Containermodells wird durch die Systemkomponente „Containerserver“ übernommen. Basis des Containerservers ist eine objektorientierte Datenbank, die um Mechanismen zur Handhabung des mehrdimensionalen Entwurfsraumes, der Basisfunktionalität der Bereichsorientiertheit (Hypothese 1) sowie um Mechanismen zur Integration von Constraints (Hypothese 2) und Partialmodellen (siehe Hypothese 3) erweitert werden mußte. Die Bereichsorientiertheit erlaubt zugleich die uniforme Handhabung des in Hypothese 4 geforderten History-Mechanismus.

Als Anwendungsdomäne unseres Projektes wählten wir einen kleinen, dafür aber gut strukturierten Ausschnitt einer Architekturmethodik, nämlich den Gebäudebaukasten MIDI für mehrgeschossige sog. hochinstallierte Gebäude, wie Schulen, Bürogebäude, Laboratorien etc., von Prof. Fritz Haller [Hal80, Hal97, HMS94a]. Dieser Baukasten ist in seiner Anzahl der Baukomponenten beschränkt. Seine Anordnungsregeln sind, im Gegensatz zu weiten Teilen der Architektur, definiert und in einem Handbuch gut dokumentiert. Anhand dieses Teilbereiches der Architektur haben wir Methodiken entwickelt, wie strukturiertes Wissen der einzelnen Experten in Form von Partialmodellen und Constraints in den allgemeinen Entwurfsraum eingebracht werden kann, um den Entwurfszustand sukzessive nach Hypothese 2 zu verfeinern. Die Integration der Anwendungsdomäne MIDI zieht sich daher durch alle Schichten unserer Systemarchitektur.

2 Ergebnisse

Die Ergebnisse, die wir im einzelnen im Projekt erzielt haben, werden entlang der oben beschriebenen Systemarchitektur vorgestellt. Die erarbeiteten Konzepte auf den verschiedenen Architekturebenen wurden prototypisch implementiert. Die Implementierung erfolgte auf Unix-Basis unter Verwendung der Programmiersprachen C++ und ObjectiveC sowie dem objektorientierten Datenbanksystem ObjectStore. Die Datenhaltungsteile sind auf SUN-Workstations, die graphische Entwurfsplattform auf NEXT-Stations realisiert, so daß eine heterogene Rechnerkopplung notwendig wurde.

Als wichtigste Ergebnisse sehen wir die folgenden im Projekt entwickelten Konzepte:

- die Integration der Entwurfsdaten für den gesamten architektonischen Planungsprozeß mit dem bereichsbezogenen Containermodell als Integrationskern
- die Konzeption und Evaluation der Benutzeroberfläche mit dem A4-Raum als einheitliche graphische Entwurfsplattform
- die Fachplanerintegration über Partialmodelle und Containerkernmodell und ihre Darstellung auf der Entwurfsoberfläche
- die einheitliche Zurückführung von Konsistenzüberprüfungen, Backtracking, Anfragen an die Entwurfsdatenbasis und Entwerferinteraktionen auf die Kollision von Bereichen.

Im folgenden wird zuerst der A4-Raum und das daraus resultierende zentrale Modellierungskonzept für die Entwurfsdaten, der Container, eingeführt.

2.1 Das A4-Modell und der Begriff des Containers

Eine zentrale Rolle innerhalb derEntwicklungen des ArchE-Projektes spielt das sog. A4-Modell [Hov94]. Das A4-Modell basiert auf dem Planungsmodell des Allgemeinen Installationsmodells *armilla* [Hal85,Hal97] und weitet dieses auf andere Bereiche des Gebäudes (Räume, Tragwerk, Fassade, Wände etc.) sowie auf die anderen Phasen des Lebenszyklus (Verwaltung, Steuerung, Nutzung, Umnutzung, etc.) aus. Es verfolgt einen komponentenbasierten Ansatz und bleibt damit der Ideologie der Gebäudebaukästen verhaftet.

Kernhypothese des A4-Modells ist, daß Planer und Nutzer ein Gebäude nicht, wie üblich, in den 3 räumlichen Dimensionen planen und betreiben, sondern innerhalb eines vieldimensionalen Datenraumes. Dieser besteht aus einer beliebigen Anzahl kontinuierlicher und diskreter Achsen, die, wie die räumlichen Achsen, orthogonal zueinander stehen und so einen vieldimensionalen Raum aufspannen. Alle Elemente des Gebäudeentwurfes können auf diesen Achsen in einer einheitlichen Art und Weise eindeutig als vieldimensional räumliche Bereiche, den sog. **Containern**, abgelegt werden.

Im ArchE-Projekt wurde der Datenraum auf folgende Dimensionen spezifiziert:

x, y, z, t (Zeit), tt (timetag), Teilsystem, Morphologie, Auflösung, Größenordnung, Nutzer, Komposition.

Dadurch können die Entwurfsentscheidungen aller beteiligten Experten aus allen Phasen des Entwurfes formalisiert nach ihrem geometrischen Ort, ihrer Entwurfszeit, ihrer Montagezeit, ihrem Maßstab, ihrer Genauigkeit, ihrer Morphologie, ihrem Entwurfsaspekt, ihrem Entwerfer und ihrer Version eindeutig abgelegt und wiederaufgefunden werden. Diese Darstellung gewährleistet eine Unterscheidung der Entwurfsentscheidungen nach geometrischen, zeitlichen, maßstäblichen, kontext-, sicht- und versionsabhängigen Kriterien. Es entsteht so die Möglichkeit, den gesamten Vorgang des Gebäudeentwurfes zu erfassen, einschließlich der unscharfen Entwurfsentscheidungen, die zu Beginn des Entwurfs vorherrschend sind. Diese Darstellung nennen wir im Sinne der ersten Hypothese bereichs- bzw. vorgangsorientiert.

Als Entwurfsentscheidungen gelten alle Objekte, die während des Gebäudeentwurfsprozesses von den Experten erzeugt wurden. Dies sind sowohl architekturspezifische Objekte wie Räume, Funktionsbereiche, Bauteilgruppen oder Bauteile als auch entwurfsspezifische wie Texte, Tabellen, Bild-, Video- oder Tondokumente. Dabei haben die Objekte eine Vielzahl unterschiedlicher Repräsentationen am Rechnerarbeitsplatz, so etwa 2D, 3D, Graphik, Photo, Text, Video, Ton .... Die jeweilige Erscheinungsform ist kontextabhängig. Auch können auf einer graphischen Benutzungsoberfläche die Entwurfsobjekte in vielfältigen inhaltlichen Zusammenhängen dargestellt sein.

Mit dem deutlicheren Hervortreten der Hypothese 1 mußte der Begriff des Containers erweitert werden. Sowohl Constraints, wie in der zweiten Hypothese beschrieben, als auch der Entwurfsausschnitt eines jeden Entwerfers, werden als räumliche Bereiche im Entwurfsraum definiert und daher ebenfalls als Container dargestellt. Dabei stellt ein Entwurfsausschnitt die Sicht eines Entwerfers auf die Teilmenge der Container des A4-Raumes dar, die er zur Zeit bearbeitet.

Eine wichtige Konsequenz diese Hypothese ist, daß man mittels der Container Interaktion beschreiben kann. Container können im Datenraum interagieren, wenn sie kollidieren, d.h. wenn sich ihre vieldimensionalen räumlichen Bereiche überschneiden. Beschreibt beispielsweise ein Container den Bearbeitungsausschnitt eines Planers innerhalb des Datenraumes, so können mehrere Personen dann miteinander kommunizieren, wenn sich ihre Betrachtungsausschnitte vieldimensional räumlich überlappen.

2.2 Entwurfsplattform

Hauptcharakteristik der Entwurfsplattform ist die Umsetzung der geschilderten einheitlichen Betrachtung von Daten, Applikationen, Constraints und Entwurfsausschnitten als räumliche Bereiche und ihre Positionierung als Container in einem mehrdimensionalen Entwurfsraum.

Interaktion und Navigation aller Komponenten des Entwurfsraumes werden ebenfalls, wie im folgenden beschrieben, vorwiegend räumlich realisiert und koordiniert. Diese uniforme Betrachtungsweise aller Elemente und Aktionen des Entwurfsprozesses ermöglicht uns die Entwicklung einer durchgängigen Entwurfsplattform. Kontroll-, Manipulations-, Kommunikations- und Navigationsaufgaben können parallel und in einheitlicher Art und Weise durchgeführt werden .

Grundgedanke der Entwurfsplattform ist, daß alle Container für den Entwerfer potentiell sichtbar sein müssen, um sie dann in der Hauptsache graphisch kontrollieren und manipulieren zu können. Diese Sichtbarkeit wird durch die drei physikalisch räumlichen (x, y, z) und die zeitlichen Dimensionen der Container bestimmt. Die Benutzungsoberfläche ist die Schnittstelle zwischen dem einzelnen Entwerfer und dem Entwurfsraum. Sie dient der Kontrolle aller Vorgänge, der Interaktion mit den Containern und anderen Entwerfern, sowie der Navigation durch den Entwurfsraum.

Als Basiskomponenten der Entwurfsplattform werden im folgenden der im Projekt entwickelte geometrische Editor und der semantische Browser vorgestellt und auf einige der im Projekt realisierten Container eingegangen.

2.2.1 Der geometrische Editor M5PSEdit

Der geometrische Editor M5PSEdit bildet die geometrische Sicht (x, y) auf den Datenraum. Gemäß der Grundidee des Datenraumes ist diese geometrische Sicht die visuelle Integrationsebene für alle Vorgänge während der Planung und Nutzung von Gebäuden. Besonderheiten des Editors sind die Ellipsendarstellungen für Container und das weltweite Koordinatensystem. Die Ellipsendarstellung wird für Container verwendet, die auf der Detaillierungsstufe der Skizzen, bzw. Bereiche angeordnet werden. Diese Darstellung gewährleistet gegenüber einer üblichen rechtwinkligen Darstellung die Sichtbarkeit vieler

übereinanderliegender Container und damit ihre gegenseitige Koordination. Das weltweite Koordinatensystem realisiert die geometrische Erreichbarkeit räumlich entfernter Orte auf einer gemeinsamen Planungsebene.

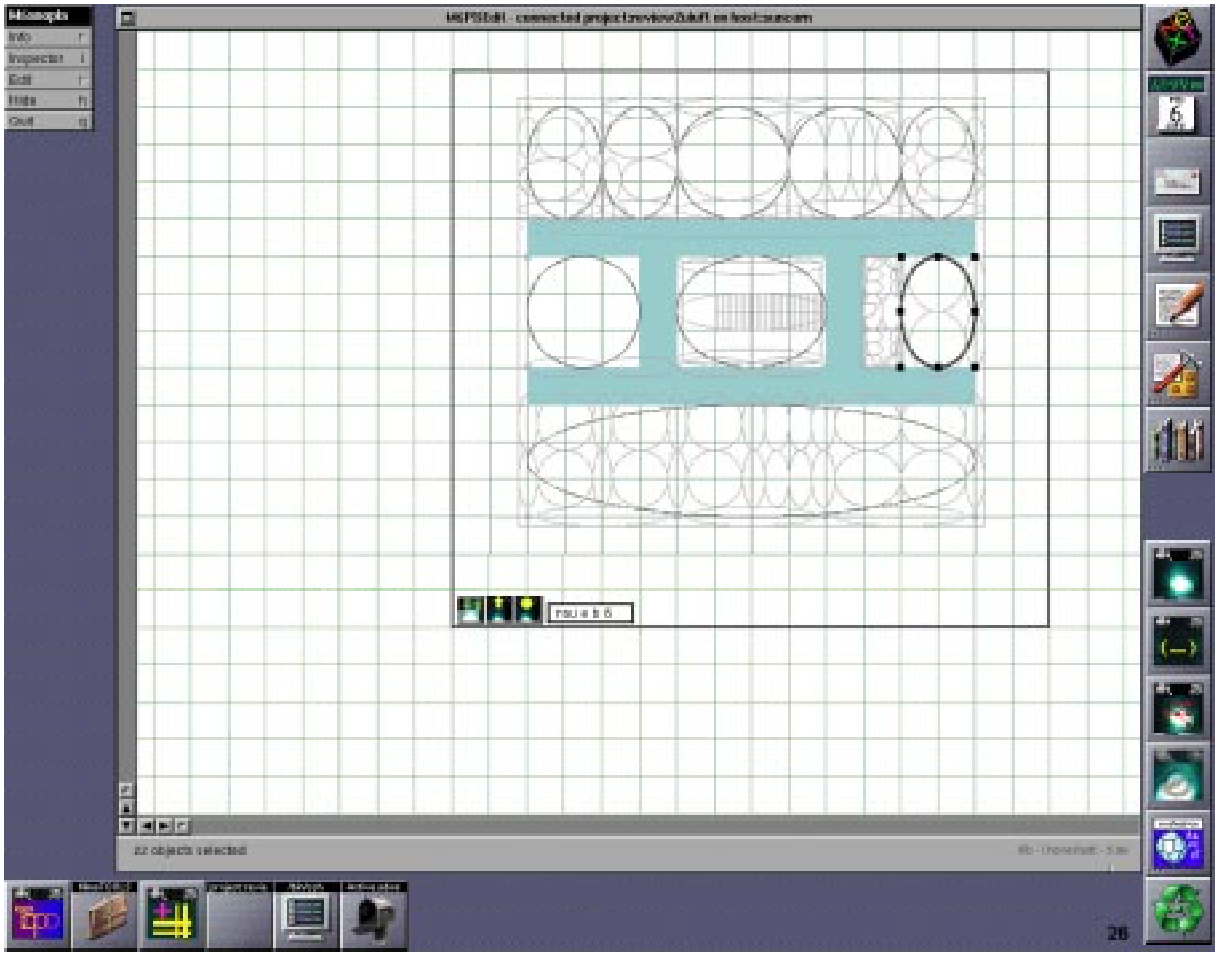


Abb. 2.1 *Der geometrischeEditor* ***M5PSEdit***
Dargestellt ist eine Planungssituation in derdas Raumlayout eines Bürogebäudes in der Maßordnung desGebäudebaukastens MIDI geplant wird. Die Planung befindet sich auf derAbstraktionsstufe der Bereiche. Es werden Räume und Nutzungbereichedefiniert. Ein Raum ist selektiert.

Der geometrische Editor M5PSEdit wurde mit Grundfunktionalitäten zur Erzeugung, Manipulation und Visualisierung von Containern ausgestattet. Weitere Funktionalität wurde über eine entsprechende Schnittstelle (M5DataIE) zu kommerziellen Editoren an die Planungsoberfläche angebunden. Es handelt sich dabei um CAD-Funktionalität durch die Systeme MiniCAD und AutoCAD, um die dreidimensionale interaktive Visualisierung des Datenraumes durch die VR-Software VRT-Superscape sowie um das Erzeugen fotorealistischer Bilder durch die Render-Software Alias-Wavefront.

2.2.2 Der semanitsche Editor M5Browser

Der semantische Editor M5Browser bildet die semantische Sicht auf den Datenraum. Er realisiert den Zugriff und die Manipulation der Container über einen Browser. In ihm können alle, insbesondere die nicht-geometrischen Dimensionen sowie die Beschreibungsattribute der Container angezeigt und editiert werden. Der semantische Editor realisiert damit den Zugriff auf alle im Containermodell modellierten Attribute. In Fortführung wären weitere Editoren vorstellbar, die den Zugriff auf die Attribute der angegliederten Fachmodelle gewährleisten.

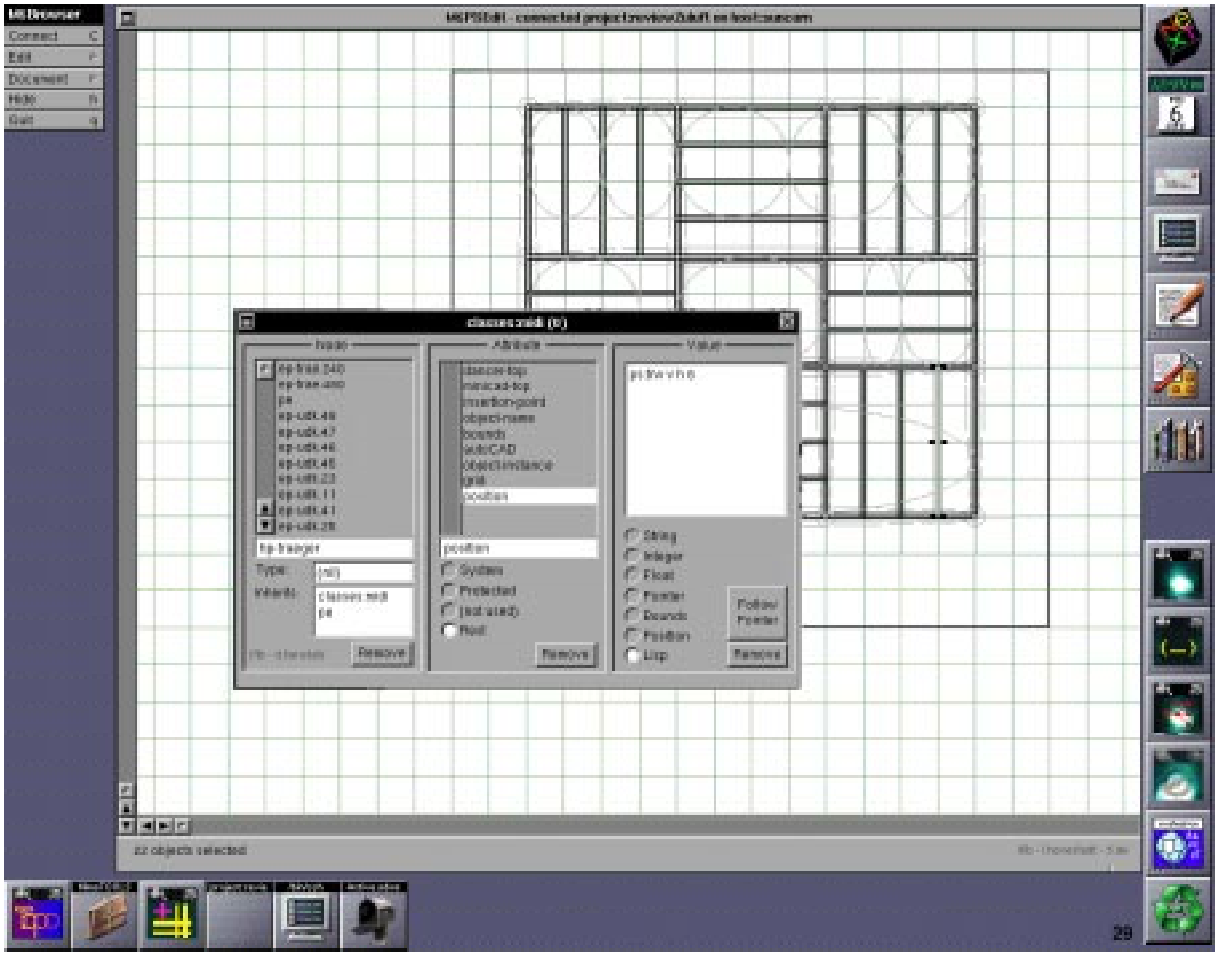


Abb. 2.2 *Der semantischeEditor* ***M5Browser***
Auf der Grundlage des Raumlayouts wird das MIDI-Tragwerk auf der Abstraktionsstufe der Hüllen geplant. In diese Planung ist das Interface des semantischen Editors eingeblendet. Dieser Editor erlaubt die Eingabe und Manipulation aller Dimensions- und Beschreibungsattribute der im geometrischen Editor selektierten Container.

2.2.3 Implementierte Container

In ArchE werden passive und aktive Container unterstützt. Die Unterscheidung ergibt sich durch ein unterschiedliches Maß an Funktionalität, das die Container innerhalb ihres mehrdimensional-räumlichen Bereiches, der durch die Dimensionattribute beschrieben wird, integrieren, bzw. verkapseln.

Passive Container haben keine eigene Funktionalität. Sie repräsentieren beispielsweise die Gebäudedaten im Datenraum, wie Räume, Träger oder Möbel, aber auch andere gebäude- bzw. entwurfsrelevante Information, wie Grafiken, Texte, Tabellen, Bilder oder Gespräche, in den unterschiedlichen Sichten, Abstraktionen und Größenordnungen. Die Darstellung dieser Daten auf der Oberfläche wird durch die Beschreibungsattribute bestimmt. Ein Container kann mehrere dieser Attribute, wie Grafik-, Text-, Bild- oder Tonattribute, besitzen und damit sichtabhängig in unterschiedliche Zusammenhänge eingebunden werden.

Aktive Container haben eigene Funkionalität. Bei Bedarf wird diese Funktionalität, bzw. die Module, die individuell benötigt werden, in den graphischen Editor geladen oder falls es sich

um eine komplexe Funktionalität handelt, die nicht vollständig in den graphischen Editor geladen werden sollte, wird diese über Ein- undAusgabeschnittstellen repräsentiert.

Die Beispielplanung in Abb. 2.3 befindet sich auf der Detaillierungsstufe der Hüllenplanung. In die Planung sind zwei Navigatoren als Beispiele für aktive Container eingeblendet, die zwei Betrachtungsausschnitte auf den Datenraum definieren. Im Hintergrund legt ein Navigator die Sicht auf die Hüllenplanung des Tragwerks (Tragwerk Verbindung Hülle nil), im Vordergrund auf die Hüllenplanung der Abluft (Abluft Verbindung Hüllenil). Der letztgenannte Navigator ist selektiert und mit seinem Benutzungsinterface eingeblendet.

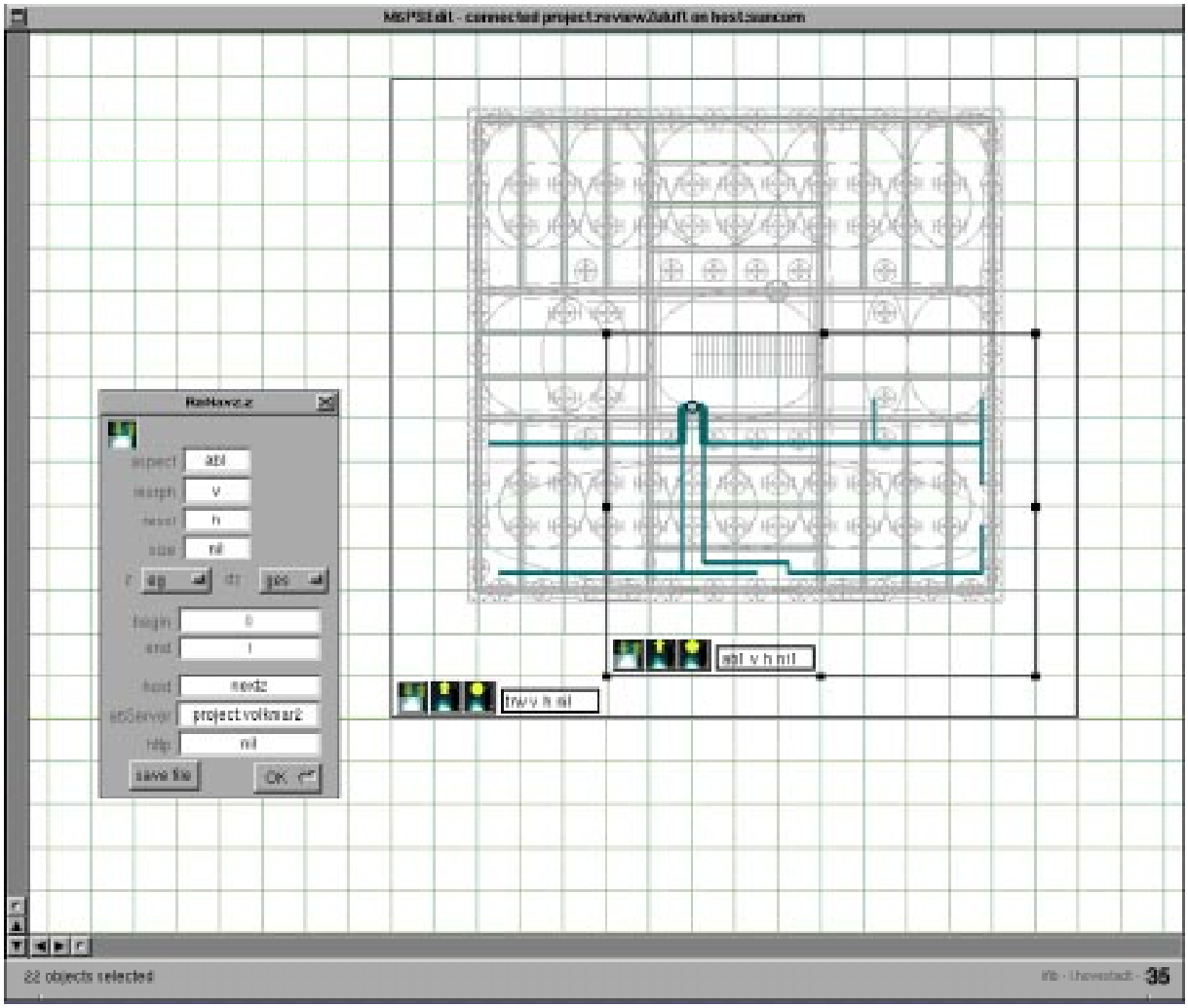


Abb. 2.3 *Beispiel eines aktiven Containers: Der Container „Navigator“*

Die Navigation erfolgt, indem sich der Navigator mit seinem räumlichen Bereich, dem Navigationsbereich, mit dem Datenraum in Kollision bringt. Der Navigationsbereich ergibt sich durch die geometrischen Dimensionen x und y, die der Navigator im Planungskontext einnimmt, sowie die weiteren Dimensionsattribute z, t, tt, aspect, morphology, resolution und size, die in seinem Benutzungsinterface eingestellt werden können.

Eine weitere Funktionalität des Navigators ist das bereichsorientierte Rücksetzen der Planung. Alle Container werden bei ihrer Erzeugung mit einem Anfangszeitstempel und bei ihrer Manipulation mit einem Endzeitstempel versehen. Alle Container der Kollisionsmenge des Navigators können also auf einen beliebigen früheren Zeitpunkt auf der timetag-Achse zurückgesetzt werden. Hintergrund dieser Funktionalität ist die Forderung, auf die hohe Dynamik innerhalb des architektonischen Entwurfsprozesses mit seinen Iterationen mit entsprechenden Konzepten zu reagieren. In [Stur97] ist die Problematik der Konsistenz-

sicherung zwischen verschiedenen Bereichen unterschiedlicher Bearbeitungstiefe und von Rücksetzmechanismen innerhalb des Datenraumes genauer behandelt.

Nach dem gleichen Prinzip der räumlichen Kollision arbeiten weitere im Projekt ArchE implementierten aktiven Container. Eine Text-, eine Mail- und eine 3D-Visualisierungsapplikation können als Container unmittelbar in den Planungskontext eingefügt werden. Sie bearbeiten jeweils die Container, bzw. ihre Attribute, mit denen sie kollidieren. Diese Form der visuellen Programmierung von Applikationen fügt sich ideal in die Semantik des Datenraumes ein und erweitert seine Leistungsfäghigkeit als Plattform für den architektonischen Planungsprozeß.

Weitere aktive Container mit einer komplexeren Funktionalität liegen z.B. in Form von Constraints für die Abwasserplanung vor. Nur ein Teil der Funktionalität, wie die Benutzungsschnittstellen zur Eingabe der Constraintsprache und zur Ausgabe der Meldungen bei Constraintverletzung werden in den graphischen Editor geladen. Die komplexe Funktionalität der Regelabarbeitung und Konsistenzsicherung ist in der ArchE-Datenbank lokalisiert.

2.3 Interaktion und Navigation

Das Grundmuster der Beziehungen zwischen den Containern ist räumlicher Natur. Es besteht aus Nachbarschaften und Kollisionen und wird durch die Position der Container im Entwurfsraum bestimmt. Nachbarschaften und Kollisionen sind, vereinfacht gesagt, räumliche Angrenzungen und Überlappungen der Container im Entwurfsraum.

Interaktions-Funktionalität, die mit Kollisionen realisiert wird, ist die Navigation eines Planers oder Nutzers durch den Datenraum: Der Betrachtungs- bzw. Bearbeitungsausschnitt eines Planers innerhalb des Datenraumes wird als Container, dem sog. Navigator, dargestellt. Ein Planer navigiert durch den Datenraum, indem er die Position seines Navigators verändert und ihn mit den anderen Containern des Datenraumes in Kollision bringt. Die Kollisionsmenge bestimmt die Menge der Container, mit der der Planer interagieren kann, und damit seine Sicht auf den Datenraum. Dem entspricht auf der Entwerferebene eine Bewegung des Entwurfsausschnitts innerhalb des mehrdimensionalen Entwurfsraumes. Mit der Bewegung wird automatisch die veränderte Position des Entwurfsauschnittes mit den Containern des Entwurfsraumes in Kollision gebracht und so die Menge der für den Entwerfer sichtbaren Container aktualisiert.

Die Interaktion mehrerer Entwerfer wird ebenfalls über Kollisionen geregelt. Mehrere Entwerfer können nur dann miteinander kommunizieren, wenn sich ihre Entwurfsräume überlappen. Dieses bereichsdynamische Konzept unterstützt die Vielschichtigkeit und Parallelität der Entwurfsaktivitäten gerade in den ersten Phasen des kooperativen Entwurfes.

Eine weitere Form der Interaktion mit den Containern ist die Formulierung, Aktivierung und Deaktivierung von Constraints. Constraints dienen dazu, wie in der zweiten Hypothese formuliert, die räumliche Modellierung des A4-Raumes zu ergänzen und den Entwurf semantisch zu strukturieren. Durch Einbringen von Constraints in Container können sie an Kollisionen teilhaben. Beispielsweise können so bei Navigation die dann gültigen Constraints ermittelt werden, und auch deren Prüfung kann automatisch angestoßen werden.

2.4 Containermodell

Das A4-Modell wurde im Projekt ArchE als Containermodell implementiert. Es umfaßt eine einzige Klasse, die Klasse Container. Die Attribute dieser Klasse unterteilen sich in Dimensions- und Beschreibungsattribute. Die Dimensionsattribute bezeichnen die im A4-Modell beschriebenen Dimensionen des Datenraumes. Die Beschreibungsattribute bezeichnen

zusätzliche Attribute, die einem Container angefügt werden können und diesen genauer, bezüglich seiner Darstellung, seines Inhalts o.ä., spezifizieren.

Die Interaktion und Navigation von den Containern des Containermodells erfolgt wie oben beschrieben auf dieser einheitlichen Basis über Kollisionen. Die Funktionalität der Container kann damit auch besonders unterstützt und optimiert werden.

2.5 Containerserver

Die Realisierung des Containermodells durch die Datenbank ist Aufgabe des Containerservers, der die Integrationsplattform für die Experten darstellt. Um dieser Aufgabe gerecht zu werden, muß der Containerserver in der Lage sein, frühzeitig Konflikte, sowohl was das gleichzeitige Entwerfen eines Ausschnittes durch verschiedene Entwerfer als auch was die Sicherstellung der zu überwachenden Bedingungen betrifft, zu erkennen. Dabei gibt es gerade in den ersten Phasen des Entwurfes viele Entwurfsbereiche unterschiedlicher Bearbeitungstiefe, so daß Inkonsistenzen zwischen diesen Bereichen eher die Regel als eine Ausnahme darstellen [Hen94]. Diese Toleranz von Inkonsistenzen sowie die geforderte bereichsdynamische Rücksetzbarkeit von Entwurfsentscheidungen aus Hypothese 3 stellen besondere Herausforderungen für eine konsistente Datenhaltung dar.

Im folgenden wird zunächst die in unserem Projekt entwickelte strukturelle Abbildung des Containermodells auf eine objekt-orientierte Datenbank vorgestellt, die eine persistente Datenhaltung der Container ermöglicht. Anschließend werden die von uns entwickelten Bereichsregeln vorgestellt.

2.5.1 Strukturelle Abbildung des Containermodells

Ein gewichtiges Problem bei der Unterstützung des gesamten Entwurfsprozesses ist der Übergang der schwach strukturierten Daten, wie sie für die frühen Entwurfsphasen charakteristisch sind, zu den stark strukturierten Daten zum Ende des Entwurfes. Die Kernidee unserer konzeptuellen Modellierung des Gesamtsystems ist die Abbildung der Containerdaten auf eine duale Darstellung der Entwurfsdaten im Containerserver. Diese setzt sich aus dem Containerkern und verschiedenen Partialmodellen zusammen. Der Containerkern beinhaltet dabei die Dimensionsattribute der Container und einen Teil der Beschreibungs- und Funktionsattribute, die für alle Entwurfsentscheidungen charakteristisch sind. Der Übergang von schwach strukturierten zu den stark strukturierten Daten wird durch den Einsatz von Partialmodellen umgesetzt. Diese erlauben fachspezifisches Wissen mit den entsprechenden Datentypen zu erfassen. Der Zusammenhang zwischen den Kerndaten und den Partialmodellen wird über die Beschreibungsattribute hergestellt. Jedem Experten sind ein oder mehrere Partialmodelle zugeordnet, die im Falle seiner Einbeziehung in den Entwurf mit in das Containermodell eingebracht werden.

In unserer Anwendungsdomäne wurde das strukturierte Wissen über den Gebäudebaukasten MIDI im Partialmodell MIDI modelliert. Entwurfsentscheidungen können damit durchgängig im Laufe des Entwurfsprozesses von unscharfen Beschreibungen aus den architektonischen Frühphasen in MIDI-Elemente überführt werden.

Gerade auf die Daten des Containerkernes wird sehr häufig zugegriffen. Hier wird die zentrale Kollisionserkennung für die Interaktion der Container sowie die Navigation der Planer durch den Entwurfsraum realisiert. Dabei ist die Kollision je nach Art der Dimension als Überlappung oder Gleichheit der Dimensionswerte der Container definiert. Die Kollisionserkennung wird durch mehrdimensionale Zugriffspfade unterstützt. Implementiert wurden der R^*-Baum, mit dessen Hilfe der Zugriff auf die kontinuierlichen Dimensionen, d.h. die geometrischen und zeitlichen Dimensionen, verbessert wird, sowie ein eigenentwickeltes, problemangepaßtes Verfahren, das den mehrdimensionalen Zugriff über die diskreten Dimensionen des

Entwurfsraumes wie Maßstab, Benutzer, Entwurfsaspekt und Genauigkeit unterstützt [Bus93, Dit95].

Aus Hypothese 3 ergibt sich die Forderung nach einem History-Mechanismus, d.h. nach Rücksetzbarkeit von Entwurfsentscheidungen aus beliebigen Ausschnitten des Entwurfs auf beliebige Zeitpunkte in der Vergangenheit. Aufbauend auf dem History-Attribut wird bei jeder Änderung eines Containers konzeptuell, d.h. von der Benutzersicht her, ein neuer Container angelegt. Intern wird Redundanz vermieden, indem nur veränderte Teile abgespeichert werden.

2.5.2 Bereichsdynamische Constraints

Nach unserer zweiten Hypothese sind Constraints notwendig, um Entwurfsentscheidungen zu formulieren und so den Entwurfsraum in Richtung auf das Zielprodukt hin einzuschränken. Zusätzlich müssen sie in der Lage sein, die Konsistenz zwischen und innerhalb von Partialmodellen der verschiedenen Experten sicherzustellen. Aus diesen Aufgaben und den Eigenarten des architektonischen Entwurfsprozesses ergeben sich folgende Herausforderungen an eine Komponente zur Handhabung der Constraints:

- Constraints müssen bereichsorientiert sein: Jedes Constraint kann in einem anderen Ausschnitt des Entwurfsraumes gelten. Ein und das selbe Constraint kann in unterschiedlichen Ausschnitten aktiviert sein. Die Constraints unterliegen hierbei ebenfalls der Dynamik des Entwurfsprozesses: Es kommen immer wieder neue Constraints hinzu, andere werden dagegen wieder ungültig oder ändern ihre Rolle in Bezug auf den Entwurf bzw. den Entwurfsausschnitt.
- Da es sich beim architektonischen Entwurfsprozeß um einen iterativen Prozeß handelt, sind bereichsbasierte Rücksetzmechanismen notwendig: Der Entwurf wird lokal in einem bestimmten Ausschnitt des Entwurfes zurückgesetzt. Dementsprechend muß in diesem Bereich ebenfalls der entsprechende ursprüngliche Zustand bezüglich der Constraints wiederhergestellt werden.
- Inkonsistenzen sind im architektonischen Entwurfsprozeß eher der Normalfall als die Ausnahme: dies ist vom Benutzer gewollt. Inkonsistenzen müssen zwar vom System erkannt werden, die Kontrolle darüber, welche Auswirkungen dies auf den Entwurf hat, muß jedoch benutzergesteuert sein in der Form, daß der Benutzer die Aktionen, die auf eine Verletzung folgen, selbst festlegen kann. Zusätzlich muß die Möglichkeit vorhanden sein, aufgrund von spezifischen Ereignissen eine Überprüfung auslösen zu können. Jedes Constraint kann hier in unterschiedlichen Ausschnitten des Entwurfs ein unterschiedliches Verhalten aufweisen, das sich in jeweils zugeordneten Ereignissen und Aktionen manifestiert.

Wir haben ein Konzept entwickelt, das diese Anforderungen erfüllt [Stur97, SML97]. Es beruht darauf, daß Constraints ebenfalls als Container im Entwurfsraum positioniert werden. Dabei kann derselbe Constraint durchaus in mehreren Containern auftreten und dort auch unterschiedliche Auswirkungen aufweisen. Die Gültigkeit der Constraints ist also an einen Bereich des Entwurfsraumes gebunden. Zusammen mit dem Verhalten, das sich in Ereignissen und Aktionen ausdrückt, sprechen wir von Bereichsregeln.

Aufbauend auf dieser Modellierung der Constraints werden eine Reihe von Mechanismen benötigt, die die oben geforderte Funktionalität sicherstellen. Aufgrund der gleichen Behandlung von Constraints und physikalischen Entwurfsentscheidungen werden zur Handhabung der Constraints ebenfalls die oben eingeführten Navigations- und Interaktionstechniken benutzt. Ein Benutzer kann so in jedem beliebigen Ausschnitt des Entwurfs die in diesem Bereich aktivierten Constraints einschließlich ihres Verhaltens und Zustands bezüglich der Konsistenz kontrollieren und sie gemäß seinen Wünschen manipulieren. Aufgrund der Forderung der Tolerierung von Inkonsistenzen erfolgt in unserem Projekt eine vollständige Entkopplung des Konsistenzbegriffes von Transaktionen. Der Benutzer spezifiziert selbst, wann Constraints zu überprüfen sind und welche Reaktionen bei Konsistenzverletzungen erfolgen sollen.

Das iterative Verhalten des Entwurfsprozesses wird durch einen Rücksetzmechanismus unterstützt, der auf Basis der Historie von Entwurfsentscheidungen arbeitet. Die Historie einer Bereichsregel beschreibt das Intervall, in dem ein Constraint unverändert blieb. Sie erfaßt demnach die Lebenszeit einer Bereichsregel. Jede Änderung bewirkt das Generieren einer neuen Bereichsregel. Es entsteht eine feingranulare Verwaltung der Constraints. Beim Zurücksetzen spezifiziert der Benutzer an der Oberfläche den Entwurfsausschnitt und den Zeitpunkt in der Vergangenheit, auf den zurückgesetzt werden soll. Nun werden alle Bereichsregeln, die zu diesem Zeitpunkt in dem spezifizierten Bereich aktiv waren, mit Hilfe der Navigationskonzepte wieder aus der Datenbank ausgelesen und erneut aktiviert. Da der Entwurfsausschnitt, der zurückgesetzt werden soll, i.d.R. nicht deckungsgleich mit den betroffenen Gültigkeitsbereichen der Entwurfsentscheidungen ist, mußten Heuristiken für eine recht komplexe Art der Rücksetzung entwickelt werden [Han95].

Das Ausführungsmodell der Bereichsregeln beruht nun auf zwei verschiedenen sich ergänzenden Ereignisbegriffen. Herkömmlicherweise wird beim Eintritt bestimmter Ereignisse eine bestimmte Menge von Constraints überprüft. Diese Ereignisse sind explizit beschrieben. Werden Ereignisse ausgelöst, so findet die entsprechende Überprüfung statt. Diese Art der Überprüfung setzen wir auch in unserem Projekt ein, nämlich dann, wenn z.B. aufgrund des Eintritts eines Meilensteines bestimmte Mengen von Constraints, die mit diesem Meilenstein verknüpft sind, überprüft werden müssen. Ergänzend zu diesem Ereignisbegriff wird ein zweiter Ereignisbegriff (sog. Bereichsereignisse) eingeführt: Wird in einem bestimmten Bereich des Entwurfs eine Änderung durchgeführt, so müssen alle Bereichsregeln, die potentiell von dieser Änderung betroffen sind, überwacht werden. Diese Entscheidung wird analog zu den Navigations- und Interaktionsmechanismen im Entwurfsraum über Kollisionsprüfung aller Bereichsregeln mit dem Bereich der Änderung getroffen.

Im Zuge der Validierung innerhalb der Anwendungsdomäne wurde zunächst eine Sammlung von MIDI-Constraints aufgestellt. Diese können, wenn der Entwerfer sich in der MIDI-Domäne bewegt, von der Oberfläche aus wahlweise in den Entwurf eingebracht werden. Darüberhinaus wurde eine erste Constraintsprache [HMS95] für ein möglichst breites Spektrum allgemeiner geometrischer Constraints entwickelt. Mit dieser Sprache bekommt der Entwerfer ein Werkzeug in die Hand, um eigene Randbedingungen, die die räumlichen Beziehungen der Container innerhalb des Entwurfsraumes beschreiben, zu formulieren.

2.6 Fachplanerintegration

Das strukturierte Fachwissen der am Gebäudeentwurf beteiligten Experten wird in Partialmodellen formalisiert. Dabei können auch Ausschnitte der Wissensdomänen modelliert werden. Diese bezeichnen wir als „Fachaspekte“ [HMS94b]. Die Integration dieser Partialmodelle in das Containermodell, die sog. Fachplanerintegration, führt zu einer Verfeinerung und Strukturierung des Entwurfes und erfolgt durch die gemeinsame Datenstruktur des Containerkerns. Die Integration erfolgt in räumlichen Bereichen. Diese bestimmen den aktuellen Entwurfs- und Gültigkeitsbereich des Partialmodelles innerhalb des mehrdimensionalen Entwurfsraums. In [Hen95] wird eine Vorgehensweise zur Integration von Fachplanern entwickelt, die einerseits die Einbindung in die graphische Entwurfsoberfläche und andererseits die Umsetzung in die Datenmodellierung beinhaltet.

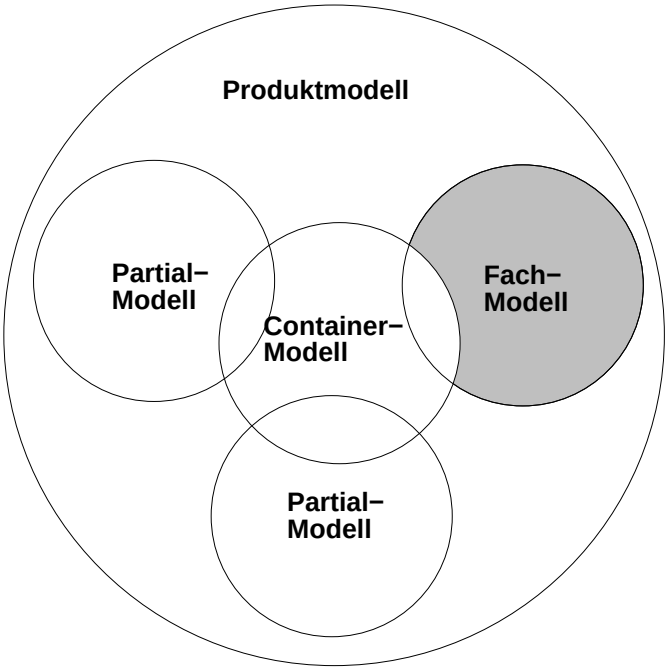


Abb. 2.3 *Das Containermodell als Integrationskern des Gebäudemodells*
Fachmodelle werden im Laufe des Entwurfes an das Containermodell angelagert. Die individuelle Verknüpfung von Containermodell und Fachmodell werden in ArchE Partialmodelle genannt.

Die Verbindung von Containermodell zu Partialmodell geschieht auf der Grundlage der Beschreibungsattribute. Wird von der Oberfläche aus auf einen Container zugegriffen, der einem bestimmten Fachaspekt zugeordnet ist, so werden die Daten aus dem Containerkern um die entsprechenden Informationen aus dem Partialmodell angereichert [Fus94]. Voraussetzung für die Ergänzung des Containers um diese Information ist dabei, daß der betreffende Fachaspekt im entsprechenden Entwurfsausschnitt aktiv ist. Wird der Fachaspekt deaktiviert, so verlieren die betroffenen Container den Zugriff auf die Information aus dem Partialmodell.

Werden mehrere Fachaspekte in den Entwurfskontext eingebracht, so kann es zu Konflikten zwischen ihnen kommen, etwa wenn zwei Experten dieselben Entwurfsgegenstände manipulieren. Diese Widersprüche werden mit Hilfe von Constraints verwaltet, wobei wir von einer evolutionären Erweiterung der Menge der Constraints ausgehen: immer wenn ein Fehler durch einen Konflikt zwischen Fachaspekten auftritt, wird ein entsprechendes Constraint ergänzt, so daß mit der Zeit ein stabiles System entsteht.

Nur ein Teil des Fachwissens der Experten läßt sich direkt in den Partialmodellen erfassen. Ergänzend wird der nicht modellierbare Teil in Constraints formuliert. Ein Zuschalten der Fachaspekte in bestimmten Ausschnitten des Entwurfes bedeutet daher auch immer ein Aktivieren der entsprechenden Constraints in diesem Ausschnitt. Dieses Aktivieren bedeutet gleichzeitig, daß, bezogen auf diesen Ausschnitt, die neu hinzugekommenen Constraints zu Beginn überprüft werden müssen. Später werden sie lediglich im Rahmen des oben angesprochenen Ausführungsmodells getestet. Das Abschalten des Fachaspekts bewirkt eine Deaktivierung dieser Constraints.

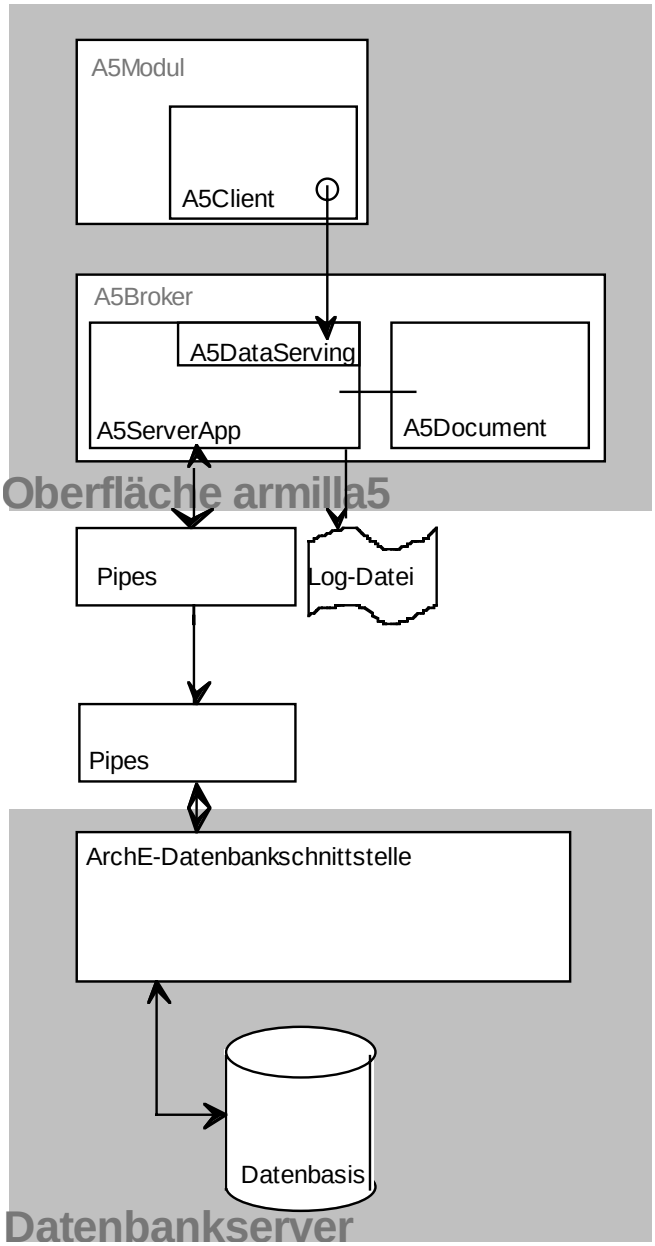
Im Vorhaben wurde exemplarisch der Gebäudebaukasten MIDI als Partialmodell modelliert und in die Entwurfsplattform als Fachaspekt eingebunden. Für unsere Anwendungsdomäne wurde zunächst das strukturelle „Wissen“ für den Gebäudebaukasten MIDI in einem konzeptuellen Schema modelliert. Darüberhinaus wurde eine Liste von Fachconstraints aufgestellt, die die Anordnungsregeln des Baukastens beschreiben. Da diese hauptsächlich durch geometrische Bedingungen beschreibbar sind, wurde im ersten Prototyp zunächst die Umsetzung dieser Bedingungen in Constraints vorgenommen [Zür96]. Es ist nun möglich, durch Zuschalten des Fachaspektes MIDI die Überführung des allgemeinen Entwurfs in einen MIDI-Entwurf durch

die Aktivierung der MIDI-Constraints und die Bereitstellung des entsprechenden Partialmodells zu unterstützen.

Des weiteren wurden im Zusammenhang mit Constraints erste Versuche zur Erfassung der Unschärfe unternommen. Auf der Modellierungsebene wird durch die Auflösungsdimension des Entwurfsraumes erlaubt, den Constraints unscharfe Gültigkeitsbereiche zuzuweisen. Unschärfe von Entwurfsinformationen läßt sich auch durch die Möglichkeit verschiedener Reaktionen auf Konsistenzverletzungen erfassen.

2.7 Prototypische Implementierung

Die nachfolgende Abbildung zeigt die Implementationsstruktur des ArchE-Prototypen. Deutlich ist die Zweiteilung in eine objektorientierte Datenbank auf einem Unix-Rechner (untere Hälfte) und eine graphische Benutzungsoberfläche unter NextStep (obere Hälfte) zu erkennen.



Die Benutzungsoberfläche besteht aus Anwendungssicht aus Programmen, die unter dem Namen "A5Module" zusammengefaßt werden und über einen gemeinsamen Kommunikations-mechanismus verfügen, der als "A5Client"-Objekt eingebunden werden kann. Die wichtigsten

A5Module sind ein graphischer Editor (M5PSEdit), ein tabellarischer Editor (M5Browser) sowie eine Kommandoschnittstelle (M5Lisp). A5Module können in beliebiger Zusammenstellung gleichzeitig betrieben werden; insbesondere sind mehrere Instanzen desselben A5Moduls möglich. Weiterhin nutzt die ArchE-Oberfläche die Möglichkeit, alle Programme auch bei beliebiger Verteilung in einem lokalen NextStep-Netz kommunizieren zu lassen. Damit ist ein verteiltes Arbeiten im selben Projekt möglich.

Das A5Client-Objekt in jedem A5Modul regelt den Zugriff des A5Moduls auf den A5Broker. Dieser arbeitet ähnlich einer Weiche zwischen den diversen Modulen und der Datenhaltung. Hierzu kann zu Testzwecken auf einen primitiven lokalen Speicherungsmechanismus zurückgegriffen werden, der bereits die Verwendung mehrere A5Broker-Instanzen mit automatischem Datenabgleich unterstützt. In der Regel wird eine externe objektorientierte Datenbank verwendet, die die Constraintdefinition und -behandlung unterstützt. Für die Datenbankverbindung werden zwei sogenannte "named pipes" verwendet, die den Zugriff auf den Kommunikationskanal ähnlich wie den auf eine Datei erlauben. Dieses Verfahren ist derzeit auf eine A5Broker-Instanz beschränkt und hat sich bei der verwendeten Betriebssystemkonstellation gegenüber Socketaufrufen als das stabilere erwiesen. Zusätzlich kann der A5Broker eine Logdatei erzeugen, in der die Kommunikationsvorgänge dokumentiert werden.

Datenbankseitig werden die ankommenden Informationen der Oberfläche in einem C++-Programm, der ArchE-Datenbankschnittstelle, aufbereitet und schließlich von der ObjectStore-Datenbank bearbeitet. Der Rückweg erfolgt analog unter Nutzung des anderen der beiden Kommunikationskanäle. Die notwendigen Protokolle für den Datenaustausch über die Kommunikationskanäle (das Pipe-Protokoll), zwischen A5Browser und A5Modulen (das A5DataServing-Protokoll), die Kommandoschnittstelle des Moduls M5Lisp und der Aufbau der Logdatei sind in projektinternen Programmdokumentationen beschrieben.

Die Datenhaltung liegt zentral auf einer SUN-Workstation unter dem Betriebssystem UNIX. Als Basis wird das kommerziell verfügbare objekt-orientierte Datenbanksystem ObjectStore eingesetzt. Darüber entstanden im Projekt entsprechende Schichten, die den Umgang mit den Containern und den angegliederten Partialmodellen besonders unterstützen. Insbesondere der Umgang mit den im 11-dimensionalen Raum organisierten Containern wird mittels bereichsorientierten Zugriffspfaden und entsprechenden Zugriffsalgorithmen unterstützt. Die bereichsorientierten Constraints werden sowohl von der Spezifikation und entsprechenden Manipulationsoperatoren als auch von der Überwachung der Constraints und Erkennung von auftretenden Constraint-Verletzungen über entsprechende im Projekt entwickelte und der ObjectStore-Datenbais vorgeschaltete Module realisiert.

3 Resümee und Ausblick

Die Ergebnisse des Projekts sind äußerst ermutigend. Das Konzept des A4-Raumes erwies sich als leistungsfähige Grundlage für eine Integrationsplattform für den Gebäudeentwurf. Auf der Entwurfsplattform wurde ein Framework für die Einbindungen multimedialer Visualierungs-techniken realisiert. Eine hyperbolische Benutzungsoberfläche ermöglicht die „unendliche“ Entwurfsfläche auf einen „endlichen“ Bildschirmausschnitt abzubilden. Durch die duale Modellierung zusammen mit den Abbildungstechniken wurde die Basis für den Übergang von den schwach strukturierten Daten, vor allem zu Beginn des Entwurfs zu den stärker strukturierten Daten der späteren Phasen erreicht. Die Constraintkomponente erlaubt die Einbindung und adäquate Handhabung des durch Constraints formulierten zusätzlichen Entwurfswissens.

Gegenüber dem Arbeitsplan mußte an einer Stelle ein veränderter Schwerpunkt gesetzt werden, der sich in der zweiten Arbeitshypothese niederschlug. Der Antrag ging davon aus, daß das Hauptproblem bei der Behandlung der Constraints in ihrer Formulierung liegt. Es stellte sich jedoch heraus, daß vorrangig vor der Problematik der Formulierung von Constraints eine Reihe von Konzepten für einen bereichsbezogenen, dynamischen Constraint-Mechanismus entwickelt

und in das Gesamtsystem integriert werden mußten. Der ursprüngliche Arbeitsplan wurde daher in dieser Hinsicht abgewandelt und ergänzt.

Das wohl bedeutsamste Ergebnis liegt in der bereichsorientierten Modellierung und Vorgehensweise, die einen Rahmen für eine orthogonale und dennoch einheitliche Behandlung zahlreicher Aspekte des Entwurfsprozesses bildet und damit für die Durchgängigkeit architektonischer Entwurfsprozesse sorgt.

Das konzipierte System wurde durchgehend in einem Prototyp implementiert, der eine gute Integrationsbasis bildet. Damit konnten die erzielten Ergebnisse für eine flexible Entwurfsumgebung, wie sie im Gebäudeentwurfsbereich gegeben sind, erfolgreich in einer ersten eingeschränkten Version validiert werden. Für die Validierung im Rahmen eines architektonischen Entwurfes wurde die Entwurfsoberfläche um Entwurfswerkzeuge erweitert.

Die Erfahrungen aus dem Vorhaben bieten Anregungen für weitere Arbeiten. So bieten die Ergebnisse eine vielversprechende Ausgangsbasis für den Nachweis der Einsetzbarkeit des Integrationsansatzes für alle Phasen des Gebäudelebenszyklus, wie das Facility Mangagement. Dazu bedarf es der konsequenten Umstellung auf eine voll verteilte Entwurfsumgebung. Die Interaktion der Container sollte um Konzepte der visuellen Programmierung erweitert werden, um den Umgang mit den verschiedenen Elementen des Entwurfes innerhalb des Entwurfsraumes wie Daten, Applikationen, Constraints und Entwurfsausschnitten, intuitiver zu gestalten und auf der graphischen Ebene wesentlich zu verbessern.

Zitate *(soweit nicht unter Veröffentlichungen aufgeführt)*

[Hal85] F. Haller:
Armilla - ein Installationsmodell. Institut für Industrielle Bauproduktion, Universität Karlsruhe, 1985

[Hal97] F. Haller:
MIDI - Armilla - Gesamtbaukasten Installationsmodell. Solothurn, Schweiz, 1997

[Hov94] L. Hovestadt:
A4 - Digitales Bauen - Ein Modell für die weitgehende Computerunterstützung von Entwurf, Konstruktion und Betrieb von Gebäuden. Fortschrittsberichte VDI, Reihe 20 Rechnerunterstützte Verfahren, Bd. 120, Düsseldorf (Dissertation am Institut für Industrielle Bauproduktion, Universität Karlsruhe), 1994

[Hal80] F. Haller: MIDI - ein offenes System für mehrgeschossige Bauten mit integrierter Medieninstallation. USM Bausysteme Haller, 1980

4 Formale Angaben

Laufzeit

3 Jahre (September 1993 - September 1996)

Beteiligte Institute

Institut für Programmstrukturen und Datenorganisation (IPD)
Institut für Industrielle Bauproduktion (IFIB)
an der Universität Karlsruhe

Projektleiter

Dr.-Ing. Peter C. Lockemann, ord. Professor (IPD)
Dr. h.c. Fritz Haller, emeritierter ord. Professor (IFIB), freier Architekt

Dr. es. sc. tech. Niklaus Kohler, ord. Professor (IFIB)

Beteiligte Mitarbeiter

Vollfinanziert (DFG):

Dr. Rose Sturm (IPD)
Volkmar Hovestadt (IFIB)

Mitarbeiter aus der Grundausstattung:

Jutta A. Mülle (IPD) zu 30%
Kay Friedrichs (IFIB) beratend
Dr. Ludger Hovestadt (IFIB) beratend
Dirk Henckels (IFIB) seit 1995

5 Veröffentlichungen

5.1 Im Projekt entstandene Veröffentlichungen

[Abr95] K. Abramowicz, B. Boss, V. Hovestadt, J. A. Mülle, R. Sturm und P. C. Lockemann: Konsistenzüberwachung in Datenbanksystemen - Eine Anforderungsanalyse anhand der Entwurfsbereiche Architektur und Schiffbau, in Datenbanken in Proc. Büro, Technik und
Wissenschaft (BTW)}, Springer Verlag, Reihe Informatik Aktuell, 1995, pp. 302-347

[Hal93] F. Haller, K. Friedrichs, V. Hovestadt, P. C. Lockemann, J. A. Mülle, R. Sturm:
The Design Navigator. Proc. 5th Intl. Conf. on Computing in Civil and Building Engineering, Anaheim, California, Computing of V-ICCCBE, 1993, pp. 335-340

[HGD95] V. Hovestadt, O. Gramberg, O. Deussen:
Hyperbolic User Interfaces for Computer Aided Architectural Design. In: Proc. CHI'95 Human Factors in Computing Systems, 7.-11.5.1995, Denver, Colorado, ACM Press, 1995.

[HH98] V. Hovestadt, L. Hovestadt: The ARMILLA Project. In: Automation in Construction. Elsevier, erscheint 1998

[HH97] L. Hovestadt, V. Hovestadt: ARMILLA5 - Supporting Design, Construction and Management of Complex Buildings. In: Proc. of CAAD futures '97, 4.-8.8.97, München, 1997

[HMS95] V. Hovestadt, J. A. Mülle, R. Sturm: Cosntraint-Sprache im Projekt ArchE. Interner Projektbericht ArchE, Universität Karlsruhe, 1995

[HHMS94] L. Hovestadt, V. Hovestadt, J. A. Mülle, R. Sturm:
ArchE - Entwicklung einer datenbankunterstützten Architektur-Entwurfsumgebung. Technischer Bericht Universität Karlsruhe, Fakultät für Informatik, Nr. 23, November 1994

[HMS94a] V. Hovestadt, J. A. Mülle, R. Sturm: MIDI-Beschreibung. Interner Projektbericht ArchE, Universität Karlsruhe, 1994

[HMS94b] V. Hovestadt, J. A. Mülle, R. Sturm: Grundlagen des ArchE-Projekts. Interner Projektbericht ArchE, Universität Karlsruhe, 1994

[LMSH95] P. C. Lockemann, J. A. Mülle, R. Sturm, V. Hovestadt:

Modeling and integrating design data from experts in a CAAD-environment. In: R.-J. Scherer (ed.): Proc. European Conference on Product and Process Modelling in the Building Industry, Dresden, A.A. Balkema, 1995, pp. 29-34

[LSMH95] P. C. Lockemann, R. Sturm, J. A. Mülle, V. Hovestadt:
Area-dependent constraints for design control in a CAAD environment. In: P.-J. Pahl, H. Werner (eds.): Proc. Intl. Conf. on Computing in Civil and Building Engineering, Berlin, A. A. Balkema, 1995, pp. 744-762

[SML97] R. Sturm, J.A. Mülle, P. C. Lockemann:
Collision of Constrained Work Spaces: A Uniform Concept for Design Interactions. Proc. 2nd Int. IFCIS Conference on Cooperative Information Systems (CoopIS), June 24-27, 1997, Charleston, South Carolina, 1997, pp. 25-35

[SML95] R. Sturm, J. A. Mülle, P. C. Lockemann:
Temporized and localized rule sets. In: T. Sellis (ed.): Proc. 2nd Intl. Workshop on Rules in Database Systems (RIDS), Athen, Springer, 1995, pp. 131-146

[SL94] R. Sturm, P. C. Lockemann:
Bereichsdynamische Konsistenzüberwachung. Technischer Bericht, Universität Karlsruhe, Fakultät für Informatik, Nr. 24, 1994

[ZBHH93] C. Ziegler, R. Bhat, L. Hovestadt, V. Hovestadt:
Digitale Räume, Installation im EVE (extended virtual environment) von Jeffrey Shaw. Multimediale III, ZKM Karlsruhe, November 1993

5.2 Im Projekt entstandene Dissertationen, Diplom- und Studienarbeiten

Dissertationen

[Stu97] Rose Sturm:
Dynamische Regelmengen zur Beschreibung von Entwurfsspielräumen. Dissertation an der Universität Karlsruhe, VDI-Verlag, Reihe 10: Informatik/Kommunikationstechnik, Nr. 495, Düsseldorf, 1997

[Hov98] Volkmar Hovestadt:
Informationsgebäude. Ein Integrationmodell für Architektur und Informationstechnologien. Dissertation am Institut für Industrielle Bauproduktion, Universität Karlsruhe, erscheint 1998

Diplomarbeiten

[Bus93] Dieter von Buschmann:
Mehrdimensionale Zugriffspfade in einer Architekturdatenbank. 1993

[Duc95] Tom Duckett:
The Elicitation and Representation of Constraints for Supporting the Workflow in an Integrated Architectural Design Environment. 1995

[Fus94] Adrian Fussek:
Konzept und Implementierung der Architektur-Datenbankschnittstelle. erweiterte Studienarbeit 1994

[Gra94] Oliver Gramberg:
Hyperbolische Benutzungsoberflächen. 1994

[Gra96] Michael Gravenhorst:
Entwicklung eines 3D-Editors für das Bausystem MIDI. 1996

[Hen95] Dirk Henckels:
Entwicklung eines Konzeptes zur Fachaspekteintegration am Beispiel von MIDI und Armilla. 1995

[Zie94] Christian Ziegler:
Digitale Baustelle, CD-ROM. 1994

Studienarbeiten

[Dit95] Gunnar Dittloff:
Implementierung eines mehrdimensionalen Zugriffspfades (R-Baum). 1995

[Gan93] Robert Ganter:
OMT, NIAM und EXPRESS - ein Vergleich objektorientierter Modellierungssprachen. 1993

[Gla96] Natalie Glaus:
Ereignisse in ArchE. 1996

[Han95] Christoph Hanselmann:
Rücksetzen von Constraints im Projekt ArchE. 1995

[Hen94] Dirk Henckels:
Analyse eines konkreten Architekturentwurfes und Validierung der Architektur-Datenbankschnittstelle. 1994

[Mac97] Thorsten Macherey:
Erweiterung der ArchE-Interrechnerkommunikationsschnittstelle. 1996

[Zür96] Dieter Zürn:
Modellierung von Konsistenzbedingungen für das Stahlbausystem MIDI. 1996

5.3 Wissenschaftliche Kontakte und Reisen

Vorträge

Volkmar Hovestadt:
Multimedia in Architectural Education, Eindhoven, 1993

Rose Sturm:
Zugriffspfadunterstützung im Projekt ArchE, Projektpräsentation in der Gruppe Prof. Hupfer, Weimar, September 1993

Rose Sturm:
Area-dependend Triggers, ActNet-Meeting Aberdeen, Juli 1994

Jutta Mülle:
Modeling and integrating design data from experts in a CAAD-environment, ECPPM Dresden, September 1994

Volkmar Hovestadt, Rose Sturm:
Das Projekt ArchE, eingeladener Vortrag, Projektvorstellung an der ETH Zürich, Lehrstuhl Prof. Schmitt, März 1995

Jutta Mülle:

Konsistenzüberwachung in Datenbanksystemen - Eine Anforderungsanalyse anhand Entwurfsbereiche Architektur und Schiffbau, BTW, Dresden, März 1995

Rose Sturm:
Area-dependent constraints for design control in a CAAD environment. Conf. on Computing in Civil and Building Engineering, Berlin, Juli 1995

Rose Sturm:
Temporized and localized rule sets. 2nd Intl. Workshop on Rules in Database systems (RIDS), Athen, September 1995

Jutta Mülle:
Workflow Support in an Architectural Design Environment. ACTNET Meeting, Karlsruhe, Januar 1996

Dirk Henckels:
ArchE- an Archictecure Development System. University of Rhode Island, Juli 1996

Peter C. Lockemann:
Collision of Constrained Work Spaces: A Uniform Concept for Design Interactions. IFCIS Conference on Cooperative Information Systems (CoopIS), Charleston, South Carolina, Juni 1997

Kontakte

Im Rahmen des Schwerpunktprogramms „Objektorientierte Modellierung in Planung und Konstruktion“ bestanden die folgenden Kontakte:

Prof. Dr. Hupfer, Hochschule für Architektur, Weimar.

Prof. Dr. Hübner, Hochschule für Architektur, Weimar.

Prof. Dr. Gehbauer, Institut für Maschinenwesen im Baubereich, Universität Karlsruhe.

Prof. Dr. Scherer, Institut für Baumechanik und Bauinformatik, TU Dresden.

Im Rahmen von FABEL (Integration von modell- und fallbasierten Entwicklungsansätzen für wissenbasierte Systeme BMFT-Verbundvorhaben 01/IW/104): GMD Sankt Augustin, BSR-Consulting München, TU Dresden, HTWK Leipzig, Uni Freiburg.

Lehrstuhl für Bauplanung und Baubetrieb, Prof. Schmitt, ETH-Zürich

ZKM Zentrum für Kunst und Medientechnologie Karlsruhe

Extern betreute Diplom- und Studienarbeiten

Paul Lang:
ENAB - Ein elektronisches Nachschlagewerk für das Bauwesen. Studienarbeit am Institut für Maschinenwesen im Baubetrieb, Universität Karlsruhe, 1994

Georg Meier:
Entwicklung eines Modells für ein Einsatzleitsystem für die Verletztentransportfahrzeuge im Ernstfall. Diplomarbeit am Institut für Maschinenwesen im Baubetrieb, Universität Karlsruhe 1995

Lars Petersson:

Objektorientierte Weiterentwicklung eines Facility - Layout - Planungsalgorithmus und dessen Einbindung in das Gesamtsystem. Studienarbeit am Institut für Maschinenwesen im Baubetrieb, Universität Karlsruhe, 1994

Tobias Rothfuchs:
Konzeption eines objektorientierten Datenmodells für die Abbildung von Organisationsformen in Unternehmen im Rahmen der Weiterentwicklung des Planungsinstrumentariums „PRISMA-Tool“. Studienarbeit am Institut für Rechneranwendung in Planung und Konstruktion, Universität Karlsruhe, 1995

Oliver Schleider:
Objektorientierte Modellierung im Bauwesen. Studienarbeit am Institut für Maschinenwesen im Baubetrieb, Universität Karlsruhe, 1995