

# Bereichsdynamische Konsistenzüberwachung

*R. Sturm, P.C. Lockemann*

Dezember 1994

*Institut für Programmstrukturen und Datenorganisation*

*Universität Karlsruhe*

*D-76128 Karlsruhe*

*e-mail: {sturm, lockemann}@ira.uka.de*

*Interner Bericht Nr.24/94, Fakultät für Informatik*

*Universität Karlsruhe*

## **Zusammenfassung**

Eine Datenbankunterstützung des integrierten architektonischen Entwurfes stellt mannigfaltige Anforderungen, die von heutigen Datenbanksystemen nicht erfüllt werden können. Insbesondere die Überwachung von Constraints stellt hier eine große Herausforderung dar. Dieser Bericht hat zum Ziel, zunächst die Anforderungen an einen Mechanismus zur Überwachung der Constraints auszuarbeiten und anschließend ein Konzept vorzustellen, das in der Lage ist diese Anforderungen zu erfüllen.

## **Abstract**

Computer-aided architectural design (CAAD) poses a number of novel challenges to database systems. Among these, in particular, is constraint management. This report starts with a characterization of the constraints arising during the architectural design process. In the second part a technical concept is introduced, which meets these challenges.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Der architektonische Entwurfsprozeß</b>	<b>1</b>
2.1	Beispielplanung . . . . .	1
2.2	Charakteristika . . . . .	5
2.2.1	Vielschichtigkeit . . . . .	5
2.2.2	Nichtlinearität . . . . .	7
2.2.3	Individualität . . . . .	8
2.3	Dynamik . . . . .	8
2.4	Bedingungen . . . . .	8
<b>3</b>	<b>Systemkonzeption</b>	<b>9</b>
3.1	Entwurfsräume . . . . .	9
3.2	Containermodell und -server . . . . .	9
3.3	A4-Raum und Containermodell . . . . .	10
3.4	Fachplanerintegration . . . . .	15
3.5	Abbildung auf Containerserver . . . . .	15
<b>4</b>	<b>Anforderungen an das Constrainthandling</b>	<b>16</b>
4.1	Eigenschaften von Constraints . . . . .	17
4.2	Benutzung von Constraints . . . . .	21
4.3	Einbringen von Constraints . . . . .	26
4.4	Überprüfung . . . . .	27
4.5	Abbildung auf den Containerserver . . . . .	30
<b>5</b>	<b>Constraintüberwachung durch den Containerserver</b>	<b>31</b>
5.1	Bereichsregeln . . . . .	32
5.2	Vorabdefinition . . . . .	34
5.3	Zustandsübergänge der Bereichsregeln . . . . .	34
5.4	Dynamik der Bereichsregeln . . . . .	34
5.5	Auswerteverfahren . . . . .	37
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>38</b>

# 1 Einleitung

Der Entwurf eines Gebäudes ist ein vielschichtiger Prozeß, welcher durch die Zusammenarbeit von vielen verschiedenen Fachplanern<sup>1</sup> geprägt ist. Das Projekt ArchE<sup>2</sup> entwickelt eine adäquate Datenbankunterstützung für die Integration dieser Fachplanern. Dabei soll weniger der Versuch unternommen werden, das vollständige Wissen der Fachplaner in der Datenbank abzubilden, als eine allgemeine Konzeption für ihre Integration zu erarbeiten. Der Schwerpunkt der Arbeit liegt insbesondere auf den frühen Phasen des architektonischen Entwurfes, sowie auf einem Ausschnitt der architektonischen Welt, nämlich dem Gebäudebaukasten MIDI und dem allgemeinen Installationsmodell Armilla. Der Gebäudebaukasten MIDI ist ein Stahlbaukasten für mehrgeschossige, sog. hochinstallierte Gebäude, wie Krankenhäuser, Labor- und Bürogebäude. Armilla ist ein allgemeines Installationsmodell, welches eine Planungsmethodik für die konfliktfreie Leitungsführung in den Deckenhohlräumen dieser hochinstallierten Gebäude darstellt. Als Leitungssysteme gibt es u.a. die verschiedenen Medien Warmwasser, Kaltwasser, Abwasser, Sprinkler, Zuluft, Abluft, Elektrizität, Gas, und Telekommunikation. Beide Systeme MIDI und Armilla sind in ihrem Aufbau weitgehend aufeinander abgestimmt und stellen einen gut strukturierten Bereich des Gebäudeplanungsprozesses dar.

Viele Restriktionen während des Entwurfsprozesses, sowohl innerhalb eines Fachplaners als auch zwischen verschiedenen Fachplanern, können in Form von Constraints ausgedrückt werden, deren Einhaltung dann eine möglichst hohe Konfliktfreiheit des Entwurfes sicherstellt. Um die Konfliktfreiheit auch technisch durchsetzen zu können, sollen diese Constraints als Konsistenzbedingungen in Datenbanken formuliert und überwacht werden. Deshalb sollen in unserem Projekt insbesondere adäquate Mittel zur Konsistenzsicherung bereitgestellt werden.

Im folgenden erläutert eine Beispielplanung die Charakteristika des architektonischen Entwurfsprozesses. Anschließend werden die in dem Beispiel dargestellten Eigenschaften zusammengefaßt und hieraus die Anforderungen an einen Mechanismus zur Einhaltung der Constraints extrahiert. Diese dienen dann als Ausgangspunkt für die vorgestellte Modellierung einer Datenbankkomponente zur Überwachung von Konsistenzbedingungen, die die aufgestellten Anforderungen erfüllt.

## 2 Der architektonische Entwurfsprozeß

Im ersten Abschnitt wird in einem Szenario ein kurzer Ausschnitt aus einem Entwurf vorgestellt. Hieraus werden dann im zweiten Abschnitt allgemeine Charakteristika des architektonischen Entwurfsprozesses abgeleitet. Die vorgestellte Beispielplanung zeigt die Komplexität des architektonischen Entwurfprozesses auf und verdeutlicht die damit verbundenen Probleme für eine Computerunterstützung.

### 2.1 Beispielplanung

Bei jedem Gebäudeentwurf sind mehrere Experten beteiligt. Diese Zusammenarbeit reicht von einer Minimalzusammensetzung für kleinere Projekte, bestehend aus dem Auftraggeber, dem Architekten, dem Statiker, dem Vermessungsingenieur, den Behörden und Bauausführenden, hin zu Planungsteams bestehend aus bis zu 70 und mehr Personen. Die Planungsleistungen werden aus einem erhöhten Kosten-, Qualitäts- und Zeitdruck zunehmend parallel ausgeführt. In unserer Beispielplanung ist dieses gleichzeitige Planen verschiedener Experten symbolisch in

---

<sup>1</sup>Unter Fachplaner verstehen wir Architekten, Statiker, Klimaplaner, usw. Es wird hierbei nicht zwischen Computerprogrammen, etwa in Form eines Kostenkalkulationsprogrammes, und realen Personen unterschieden.

<sup>2</sup>Das Projekt wird im Rahmen des DFG-Vorhabens "Datenbankunterstützte Koordinierung und Integration von Planungswerkzeugen im Baubereich" gefördert.

ein gemeinsames Bild eingearbeitet worden. Üblicherweise arbeitet jeder auf einer eigenen Kopie des Planes.

Abbildung 1 zeigt einen Ausschnitt der Beispielplanung für einen neuen naturwissenschaftlichen Trakt der Kantonsschule in Solothurn. Die Abbildung zeigt das Layout des Erdgeschosses im Maßstab 1:100, die Einbindung in die städtebauliche Umgebung, sowie einen Möblierungsvorschlag.

Die Funktionsbereiche des Entwurfes sind in Abbildung 1 durch Zahlen gekennzeichnet. Bereich 1 stellt eine Fläche für Wechselausstellungen dar. Der sich hieran anschließende Bereich 2 ist als Kommunikationszone für die Schüler gedacht, die sich hier in den Pausen oder nach der Schule aufhalten können. Durch eine Glaswand abgetrennt soll in Bereich 3 ein Lehrmittelraum entstehen, der von allen Seiten einsehbar ist und die Neugier der Schüler wecken soll. Die Bereiche 4 stellen die Toiletten dar und die Bereiche 5 die offenen Treppenhäuser. Die Räume, die mit einer 6 versehen sind, sind Fachräume und die Räume 7 die dazugehörigen Lehrmittelräume der einzelnen Lehrer. Durch die spätere Nutzung bedingt sind sehr viele Laborräume erforderlich. Diese haben hohe Anforderungen an die Leitungsführung und an die Ausstattung der einzelnen Schülerarbeitsplätze, die beispielsweise alle mit Waschbecken ausgerüstet sein müssen. Hierbei werden die Wünsche der einzelnen Lehrer berücksichtigt, die den einzelnen Räumen fest zugeordnet sind<sup>3</sup>. Die Klassen wechseln die Räume entsprechend ihres Fachunterrichtes. Der Bereich 8 stellt noch einen speziellen Lehrmittelraum für Biologie dar, der aufgrund der Wünsche eines Biologielehrers in dieser Form mit in den Entwurf aufgenommen wurde. Er benötigt Platz für präparierte Tiere, die er in diesem Raum den Schülern vorstellen möchte.

In unserem Szenario ist die Planung zu diesem Zeitpunkt keineswegs abgeschlossen. Zum einen sind noch nicht alle Teile fertig entworfen, wie z.B. die Fassade, die konkrete Gestaltung der Umgebung oder auch die doch sehr anspruchsvolle Leitungsplanung. Zum anderen ergeben sich bei fast jedem Zwischenstadium eines Entwurfes neue bzw. revidierte Anforderungen, die ebenfalls noch berücksichtigt werden müssen. Abbildung 2 stellt diese Situation dar.

- Eine neue Anforderung kommt mit dem Wunsch, einen separaten **Filmraum** einzurichten, in welchem auf einer großen Leinwand Sachverhalte plastischer dargestellt werden können als auf den kleinen Fernsehern, die in den Fachräumen zur Verfügung stehen. Der Filmraum muß für mindestens 30 Personen Platz bieten. Die Wand nach außen muß vollständig verdunkelbar sein. Zwei Türen müssen in diesen Raum führen, und eine große Leinwand muß fest installiert werden. Außerdem ist der Platz für den ebenfalls fest installierten Vorführapparat vor den Schülern geschützt zu wählen, also am besten in einem eigenen abgegrenzten Raum.
- Aufgrund neuer Zahlen über die voraussichtliche Schülerzahl in den nächsten zehn Jahren wird ein **zusätzlicher Fachraum** benötigt, der universell einsetzbar ist. Dieser muß sowohl für Chemie als auch Biologie und Physik geeignet sein und eine gewisse Mindestgröße aufweisen. Der zusätzliche Laborraum sollte möglichst neben dem Filmraum liegen, damit Versuche, die in dem Laborraum durchgeführt werden, direkt auf die Großleinwand übertragen werden können. In diesem Fall soll eine Verbindungstür die beiden Räume so verbinden, daß der Lehrer den Überblick über beide Räume behalten kann.
- Ein weiterer Architekt arbeitet an der äußeren Gestalt des **Baukörpers** und der Art und Weise, wie er in die Umgebung eingepaßt werden soll. Der Planer legt nun fest, daß an der Vorderfront und einer Seitenfront freie Flächen, beispielsweise für den Schulhof, entstehen

---

<sup>3</sup>Die individuelle Beteiligung der einzelnen Lehrer in der Planung ist eine zusätzliche Randbedingung bei dieser Schule und nicht grundsätzlich üblich. Normalerweise werden die Lehrer durch den Bauherren repräsentiert, der die Wünsche als entsprechende Anforderungen einbringt.

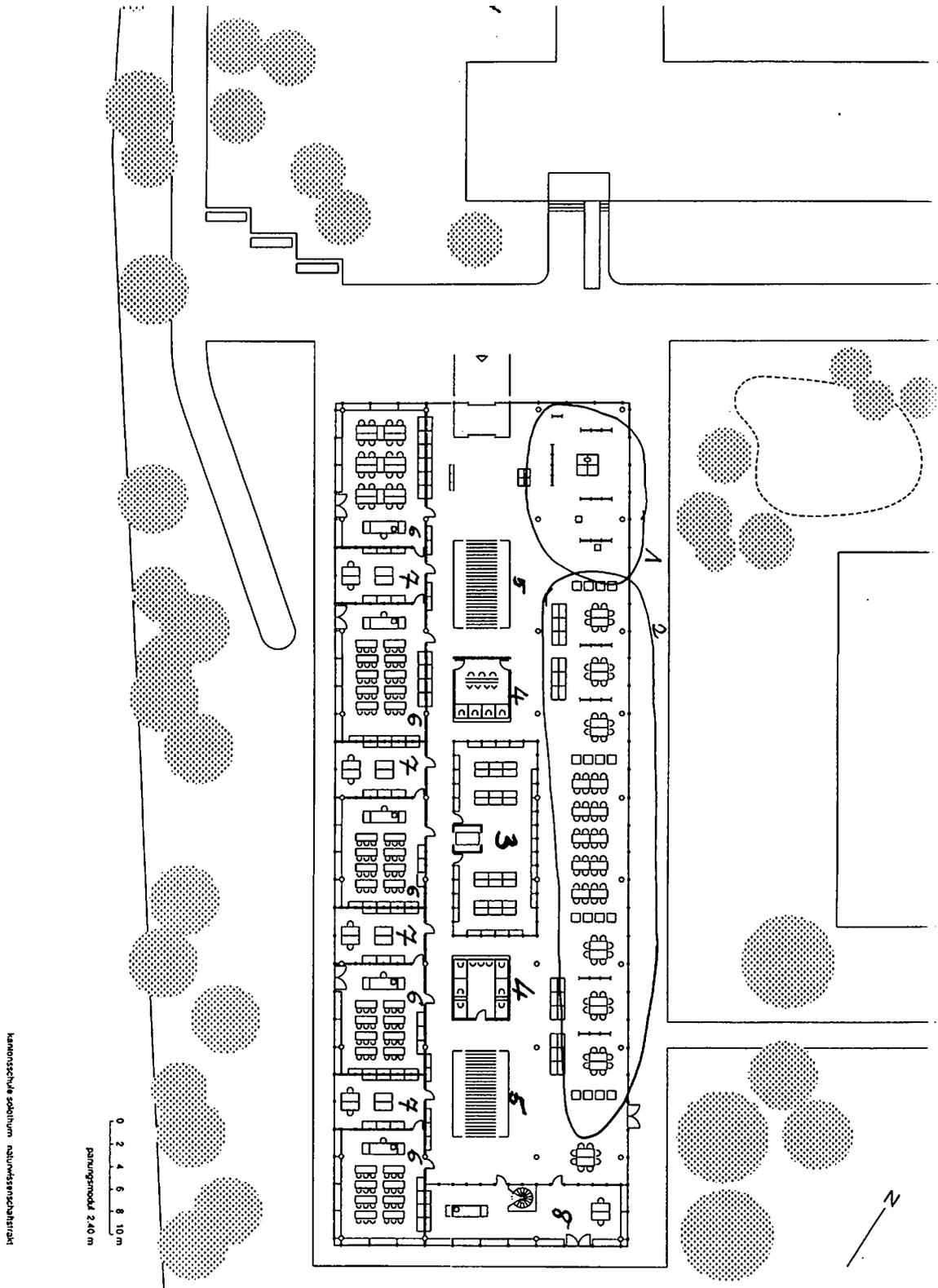


Abbildung 1: Beispielplanung: Kantonsschule Solothurn



sollen. Außerdem bringt er schon eine gewisse Vorstellung von der Gesamtfassadengestaltung in seine Skizze ein, wie dies in Punkt 2 (Abbildung 2) dargestellt ist. Das Dach soll mit Solarkollektoren ausgestattet werden, so daß ein Teil des Energiebedarfes des Gebäudes hierüber gedeckt werden kann. Als Dachform wird ein Sheddach vorgeschlagen, auf dessen sonnenzugewandten Flächen die Kollektoren angebracht werden sollen.

- Eine weitere Anforderung ist die Bereitstellung einer gewissen Anzahl von **Parkplätzen** für Lehrer. Da diese bei der alten Schule bisher noch nicht vorhanden sind, müssen neue Flächen hierfür bereitgestellt werden. Entsprechend überlegt sich der Architekt, wie diese Anforderungen am besten in die Umgebung des Gebäudes integriert werden können, und stellt seine Lösungsalternativen in dem Gesamtentwurf zur Diskussion (siehe Punkt 3, Abbildung 2).
- Der nächste Planer beschäftigt sich mit der **Detaillierung der Fassade** sowie mit der Konzeption für die **Raumentlüftung**. Für die Fassade wird an der Südseite des Gebäudes ein außenliegender Sonnenschutz vorgeschlagen (Punkt 5, Abbildung 2). Winddichtheit, Regendichtheit, und Wärmeschutz sind weitere Anforderungen, die bei der Detaillierung der Fassade berücksichtigt werden müssen. Die Raumentlüftung soll über Lüftungskanäle geschehen, die im Deckenhohlraum an der Innenseite der Fassade liegen (Punkt 4, Abbildung 2). Die Dimensionierung des Abluftsystems hängt dabei u.a. von der Größe und Nutzung der zu entlüftenden Räume ab.

Jede Änderung kann vielfältige, zwangsläufig daraus folgende Änderungen nach sich ziehen. Wird z.B. der neue Laborraum zusätzlich geplant, so müssen neue Leitungen mit den besonderen Anforderungen, die ein Laborraum stellt, geplant und in das Gesamtleitungssystem eingefügt werden. Ist damit zu rechnen, daß in diesem Laborraum chemische Versuche durchgeführt werden, so muß der Luftaustausch ungleich höher sein als in einem Raum, in dem Physik unterrichtet wird.

In Abbildung 3 ist diese Situation grob dargestellt. Die neuen Abluftleitungen müssen an bestimmten Punkten in der Decke verankert werden. Von dort muß es möglich sein, eine Astleitung<sup>4</sup> zu einer Stammleitung Klima<sup>5</sup> zu legen, wobei die angestrebte Stammleitung bisher noch nicht ausgelastet sein darf. Hierbei sind ebenfalls die vorgegebenen maximalen Längen zu berücksichtigen. Gibt es keine Möglichkeit, die Abluftleitungen im Deckenhohlraum mangels Raum zu verlegen, so führt das entweder zu einer Revision der Entwurfsentscheidung, den neuen Laborraum an diese Stelle zu planen, oder es müssen große Teile der Leitungsplanung überdacht werden, was mit hoher Wahrscheinlichkeit dann wieder Auswirkungen auf andere Teile des bisher vorliegenden Planes hat.

## 2.2 Charakteristika

Das vorangegangene Beispiel sollte einen Eindruck von dem komplexen architektonischen Entwurfsprozeß vermitteln. Analysiert man diesen Prozeß weiter ins Detail, so fallen eine Reihe von Charakteristika für den Entwurfsprozeß auf, die man als Vielschichtigkeit, Nichtlinearität und Individualität bezeichnen kann.

### 2.2.1 Vielschichtigkeit

Die Vielschichtigkeit im architektonischen Planungsprozeß bezieht sich auf verschiedene Aspekte des Entwurfs. Viele Beteiligte arbeiten gemeinsam an einem Entwurf, ein Entwurf umfaßt

---

<sup>4</sup>Leitung von einer im Stockwerk vorhandenen Verteilerstelle zu dem Nutzungspunkt.

<sup>5</sup>Zubringerleitung von der zentralen Klimaanlage in die einzelnen Stockwerke.



verschiedene Maßstabebenen oder verschiedene Genauigkeiten in der Planung. Die am Entwurfsprozeß Beteiligten - Architekt, Statiker, Klimaingenieur, Bauherr, Baubehörde, ..., - sind in unterschiedlichen, sich überlappenden Phasen am Entwurf beteiligt. Jeder der am architektonischen Entwurf beteiligten Fachplaner betrachtet die im Planungsprozeß anfallenden Daten mit seinen Augen. Dies beinhaltet eine eigene Arbeitssprache (Graphik, Semantik), eigene Vorschriften und eigene Prioritäten bezüglich des Entwurfes. Die verschiedenen **Blickwinkel** aller am Entwurf Beteiligten können als parallele Designwelten bezeichnet werden. In unserem Beispiel sind momentan gleichzeitig Architekten, Lehrer, Fassadenbauer, Statiker und Klimaspezialisten beteiligt. Durch einen erhöhten Kosten-, Qualitäts- und Zeitdruck in der Planung werden die Leistungen der einzelnen Planer zunehmend parallelisiert (Integrale Planung). Dies stellt besondere Anforderungen an die Kommunikation und die Kompatibilität der Planungsleistungen.

Die verschiedenen **Maßstabebenen** der Planung umfassen ein Spektrum vom Städtebau im Maßstab 1:1000 bis hin zum konstruktiven Detail im Maßstab 1:1. Die Maßstabebenen erfassen die verschiedenen inhaltlichen Beschreibungsebenen des Entwurfs. Der Städtebauplan beschreibt die städtebaurelevanten Einzelheiten. Er betrachtet damit naturgemäß einen größeren Ausschnitt als die Ausführungsplanung, die dafür in einem kleineren Ausschnitt den konstruktiven Zusammenhang detailliert. Die Anzahl der beschriebenen Teile pro Planausschnitt und damit die Komplexität ist in der Regel auf jeder Maßstabsebene vergleichbar. In unserer Beispielplanung arbeitet der Architekt, welcher an der Einpassung des Gebäudes in die Umgebung arbeitet, auf der städtebaulichen Ebene, während gleichzeitig andere Planer mit der konstruktiven Durcharbeitung des Entwurfes in einem wesentlich größeren Maßstab beschäftigt sind. Diese Gleichzeitigkeit stellt eine typische Herausforderung des architektonischen Entwurfsprozesses dar.

Die verschiedenen **Genauigkeiten** der Planung erstrecken sich von der vagen Skizze bis zur definierten Ausführungsplanung. Auf allen Maßstabebenen kann die dargestellte Information von vage bis exakt reichen. Die Genauigkeit kann sich also auch auf ein- und derselben Maßstabsebene mischen. Als Beispiel gibt es sowohl auf der Ebene des Städtebaus als auch auf der des konstruktiven Details unterschiedliche Genauigkeiten. Im ersten Fall reicht die Spanne von städtebaulichen Skizzen bis hin zu exakten Baulinien und Baugrenzen, im zweiten Fall reicht sie von konstruktiven Detailskizzen bis hin zu Ausführungsanweisungen mit den exakten Bauteilabmessungen.

### 2.2.2 Nichtlinearität

Ein Architekt springt typischerweise zwischen verschiedenen Ausschnitten des Planes. Es ist nicht eindeutig, welche Teile er als nächstes bearbeiten wird. Häufig werden Teile des Entwurfes zurückgesetzt. Die Vorgehensweise ist hierbei sehr stark von der einzelnen Person abhängig. Es ist beispielsweise nicht vorhersagbar, wie ein Architekt den zusätzlichen Laborraum in den Gesamtentwurf einbringt. Bis er einen konkreten Vorschlag macht, hat er in der Regel schon mehrere Alternativen gedanklich durchgespielt. Selbst nachdem er einen konkreten Vorschlag gemacht hat, ist damit zu rechnen, daß dieser wieder verworfen wird und eine neue Alternative aufkommt, häufig aufgrund der Kommunikation mit anderen am Entwurf Beteiligten.

Das Wechseln von Entwurfsausschnitten geschieht zum einen, indem auf demselben Detaillierungsgrad ein anderer geometrischer Ausschnitt ausgewählt wird, und zum anderen, indem andere Detaillierungsstufen oder andere Maßstäbe betrachtet werden. Unterschiedliche Detaillierungsstufen gibt es dabei in allen Phasen des Entwurfs. Ein Springen zwischen ihnen ist die Regel. Die Detaillierungsstufen alleine lassen noch keine Aussagen über das vorliegende Funktionsgefüge zu. Eine geometrisch vage Skizze kann schon ein festgefügtes Funktionskonzept besitzen, umgekehrt kann ein festgefügt geometrischer Raum ein noch vages Funktionskonzept haben. So hat die Skizze des Laborraumes schon ein festes Funktionskonzept, während gleichzei-

tig die Funktionalität der Fassade noch sehr unbestimmt, der geometrische Ort hingegen klar ist. Wann die Ausarbeitung des jeweils fehlenden Teiles stattfindet, ist ebenfalls nicht vorhersagbar.

Inkonsistenzen zwischen den einzelnen Entwurfsentscheidungen sind daher eher die Regel als die Ausnahme. Dennoch, trotz aller Nichtvorhersagbarkeit des Planungsergebnisses und des Lösungsweges, ist doch eine allgemeine Tendenz vom Groben zum Feinen, von der vagen Skizze zur detaillierten Ausführungsplanung feststellbar. Diese Tendenz kann über sogenannte Meilensteine, wie etwa die Phasen der HOAI<sup>6</sup>, beschrieben werden.

### 2.2.3 Individualität

Die Rahmenbedingungen des architektonischen Planungsprozesses sind im Gegensatz z.B. zum Maschinenbau nicht Serienproduktion unter klinischen Bedingungen, sondern one-of-a-kind Produktionen von prototypischen Gebäuden. Es handelt sich in der Regel um eine von der Planung bis zur Ausführung räumlich, zeitlich und inhaltlich individuelle Aufgabe. Lösungen lassen sich daher nur schwer von einem Entwurf auf den anderen übertragen. Dies wird in der Beispielplanung u.a. an der individuellen Beteiligung der Nutzer, hier in Form der Lehrer, in der Planung deutlich.

## 2.3 Dynamik

Anhand der oben dargestellten Charakteristika des architektonischen Entwurfsprozesses und des vorgestellten Beispiels wird die hochgradige Dynamik deutlich, die für einen solchen Vorgang typisch ist. Diese Dynamik hat ihren Ursprung in den folgenden Aspekten:

- Ein Architekt spielt ständig mit neuen Ideen, die alte Entwurfsentscheidungen oft rückgängig machen oder verändern.
- Der Bereich, in dem ein Architekt weiterplant, ist sehr stark von seiner Intuition abhängig und als solche nicht vorhersagbar. Auch ist ein beliebiges Springen innerhalb eines Planes eher die Regel als die Ausnahme.
- Da am Entwurf ständig viele Personen beteiligt sind, kommt es häufig zu sich widersprechenden Vorschlägen. Dies führt ebenfalls zu einer starken Dynamisierung des Entwurfsprozesses, da ein Teil der schon getroffenen Entscheidungen zwangsläufig wieder zurückgenommen und neue Alternativen gesucht werden müssen, meist mit sehr hohem Kommunikationsaufwand.
- Eine zusätzliche Dynamik entsteht dadurch, daß der Entwurfsvorgang relativ viel Zeit in Anspruch nimmt. Nicht selten kommen in diesem Zeitraum neue Anforderungen an ein Gebäude mit hinzu. Diese müssen dann mit in den Entwurfsprozeß aufgenommen werden.

## 2.4 Bedingungen

An jedem oben angeführten Punkt sind eine Fülle von neuen Bedingungen an den Entwurf gebunden, die dann im folgenden mit überwacht werden müssen. Beispielsweise fallen Bedingungen in den folgenden Gebieten an:

### 1. Bedingungen aus dem Bebauungsplan

---

<sup>6</sup>In der HOAI (= Honorarordnung für Architekten) werden Leistungsphasen beschrieben, die jede für sich, nach Abschluß einen festen Prozentsatz der Gesamtbaukosten als Honorar abwerfen. Diese Phasen wirken im Entwurfsprozeß also wie Meilensteine. Man unterscheidet : Grundlagenermittlung, Vorplanung, Entwurfsplanung, Genehmigungsplanung, Ausführungsplanung, Vergabe, Bauausführung und Objektbetreuung mit Dokumentation

2. Normen
3. Randbedingungen der einzelnen Gewerke (z.B. Gefälle der Abwasserleitungen)
4. Anforderungen des Bauherren
5. Individuelle Anforderungen der Architekten

Es treten hier nun zwei verschiedene Typen von Bedingungen auf. In die erste Gruppe fallen die Bedingungen der Punkte 1, 2 und 3. Diese Bedingungen unterliegen einem relativ festen Gefüge und verändern sich im Laufe eines Entwurfes kaum. In die zweite Gruppe fallen die Bedingungen der Punkte 4 und 5. Hier ist die Dynamik innerhalb des Entwurfsprozesses sehr hoch. Der Bauherr stellt zwar zu Beginn schon viele Anforderungen. Diese werden im Laufe des Entwurfes aber in der Regel mehrmals überarbeitet. Dasselbe gilt für die Anforderungen, die ein Architekt im Laufe der Zeit dem Entwurf hinzufügt oder modifiziert.

## 3 Systemkonzeption

### 3.1 Entwurfsräume

Ziel des Projektes ArchE, in dessen Rahmen die in diesem Bericht beschriebenen Arbeiten anzusiedeln sind, ist die Unterstützung des architektonischen Entwurfes, wie er bis hierhin beschrieben wurde. Dieser ist in den frühen Entwurfsphasen durch sehr viele Unsicherheiten und wenig Struktur gekennzeichnet. Im Laufe des Entwurfs nehmen diese Unsicherheiten dann immer weiter ab, und es bilden sich klarere Strukturen heraus. Um, trotz dieser Unterschiede, eine durchgehende Rechner- und vor allem Datenbankunterstützung zu bieten, ist ein einheitliches Modellierungs- und Verarbeitungskonzept erforderlich.

Bisherige Ansätze versuchen diese Problem gemäß der im Entwurfsbereich weit verbreiteten Integrationsmethodik zu lösen, die sich an den Gegenständen des Entwurfs, den Entwurfsobjekten, orientiert und deren Modellierungen zu integrieren versucht. Der Nachdruck der Modellierung liegt daher auf einer Vereinheitlichung der Modellbildung bei den Beteiligten, also insbesondere auch bei den Fachplanern. Die Erfahrungen zeigen, daß diese Integrationsmethodik sich bei einer sequentiellen Abfolge der Entwurfsschritte bewährt, jedoch Schwächen zeigt, wenn der Entwurf mit vielen Unwägbarkeiten, spontanen Sprüngen und Rücknahmen von Entscheidungen behaftet ist.

Unseren Arbeiten legen wir daher eine neuartige Arbeitshypothesen der Architekten zugrunde, die sich nicht an Entwurfsobjekten, sondern an Entwurfsräumen orientiert. Ein Entwurfsraum repräsentiert eine Menge von Entwurfsobjekten, die unter aktuellen Randbedingungen als zusammengehörig betrachtet werden sollen, deren Zusammenhang aber unter veränderten Randbedingungen wieder aufgelöst werden kann. Diese Arbeitshypothese scheint der Dynamik und Nichtlinearität in besonderer Weise Rechnung zu tragen. Diese Arbeitshypothese entspricht als Formalisierungskonzept dem am IFIB (Institut für Industrielle Bauproduktion) entwickelten A4-Raum ([6]). Er bietet losgelöst von jeglichen Überlegungen zur rechnergestützten Implementierung einen formalisierten Rahmen, innerhalb dessen sich die Vorstellungen des Architekten und anderer am Entwurf Beteiligter bewegen können.

### 3.2 Containermodell und -server

Der A4-Raum bietet zwei Facetten. Zum einen verleiht er der Benutzungsoberfläche und den dort ausführbaren Aktionen eine Struktur. So können auf der Oberfläche gleichzeitig verschiedene Aspekte einer Planung einheitlich dargestellt werden. Zum zweiten werden im A4-Raum

alle Planungsentscheidungen in einer einheitlichen Datenstruktur, den sogenannten Containern, repräsentiert. Deshalb sprechen wir bei diesem Entwurfsrahmen abkürzend vom “Containermodell”. Ein erster Ansatz zur Erfassung des Containermodelles findet sich in [4], in [8] ist seine Weiterentwicklung dargestellt.

Abbildung 4 zeigt die Einheitlichkeit und zugleich Vielschichtigkeit des A4-Raum-Konzeptes. Alle Container werden auf der graphischen Benutzungsoberfläche einheitlich als Rechteck mit einem spezifischen graphischen Inhalt abgebildet. Dies ist in der Regel ein 2D-Graphiksymbol, etwa eine Stütze, ein Fassadenelement, ein Straßenabschnitt oder eine Gebäudehülle. Darüberhinaus können die Container um weitere Repräsentationen, etwa eine 3D-Darstellung, eine Photographie, ein Video, oder aber auch um eine Applikation erweitert werden, die den Inhalt des Containers bearbeiten kann. Beispiele für Applikationen sind Tabellenkalkulationen, Textverarbeitung, Telefon, e-mail etc..

Die Verwaltung der Container in der Datenbank wird dabei durch den in unserem Projekt zu entwickelnden Containerserver gewährleistet. Dieser Containerserver hat folgende Aufgaben:

- Einhaltung der Bedingungen auf den Containern
- Persistenz der Container
- Ermöglichung der Kooperation verschiedener Fachplaner

Abbildung 5 zeigt das prinzipielle Zusammenspiel zwischen Containermodell und dem Server. Wir gehen in weiteren Verlauf davon aus, daß die geforderte Funktionalität über eine objektorientierte Datenbank gewährleistet wird. Ein kommerzielles System soll hierbei um die notwendige, noch nicht vorhandene Funktionalität erweitert werden.

Im folgenden werden die wesentlichen Ideen des A4-Raumes vorgestellt. Auf die von uns entwickelte Abbildung der Container im Containerserver wird anschließend eingegangen.

### 3.3 A4-Raum und Containermodell

Die Grundidee, die hinter dem A4-Raum steht, ist, daß alle Planungsentscheidungen, die in einem architektonischen Entwurf getroffen werden, in Form von Containern durch eine bestimmte Menge von Attributen eindeutig beschreibbar sind. Die Attribute werden auch Dimensionen genannt. Der Belegungsvektor einer Planungsentscheidung über alle Dimensionen hinweg wird **Position** genannt. Die Dimensionen umfassen zum einen geometrische Angaben und zum anderen Aspekte wie Zeit, Genauigkeit und Auflösung. Insgesamt unterscheiden wir 11 Dimensionen, die den A4-Raum aufspannen und eine einheitliche und eindeutige Positionierung aller Planungsentscheidungen ermöglichen.

Die Dimensionen werden im folgenden vorgestellt. Wir unterscheiden *diskrete* und *kontinuierliche* Dimensionen. Eine ausführliche Beschreibung findet sich in [7].

Kontinuierliche Dimensionen umfassen die geometrische und die zeitliche Beschreibung von Entwurfsentscheidungen. Durch die geometrische Beschreibung wird die Bounding Box des betrachteten Entwurfsobjektes dargestellt. Zwei Zeitdimensionen beschreiben die unterschiedlichen zeitlichen Aspekte von Entwurfsentscheidungen. Die erste Zeitdimension, die sogenannte Relevanzzeit, ist das Intervall, in dem ein Entwurfsobjekt für ein Gebäude von Wichtigkeit ist, also von seinem ersten Erscheinen im Entwurf bis zu dem Zeitpunkt, zu dem es später wieder aus dem realen Gebäude entfernt wird. Die zweite Zeitdimension, die sogenannte Modellierungszeit, beschreibt das Zeitintervall, in dem ein Entwurfsobjekt unverändert bleibt.

Diskrete Dimensionen beschreiben die folgenden Eigenschaften des Entwurfsobjektes:

- *Nutzer* ist der Fachplaner oder der Architekt, der für dieses Entwurfsobjekt verantwortlich ist.

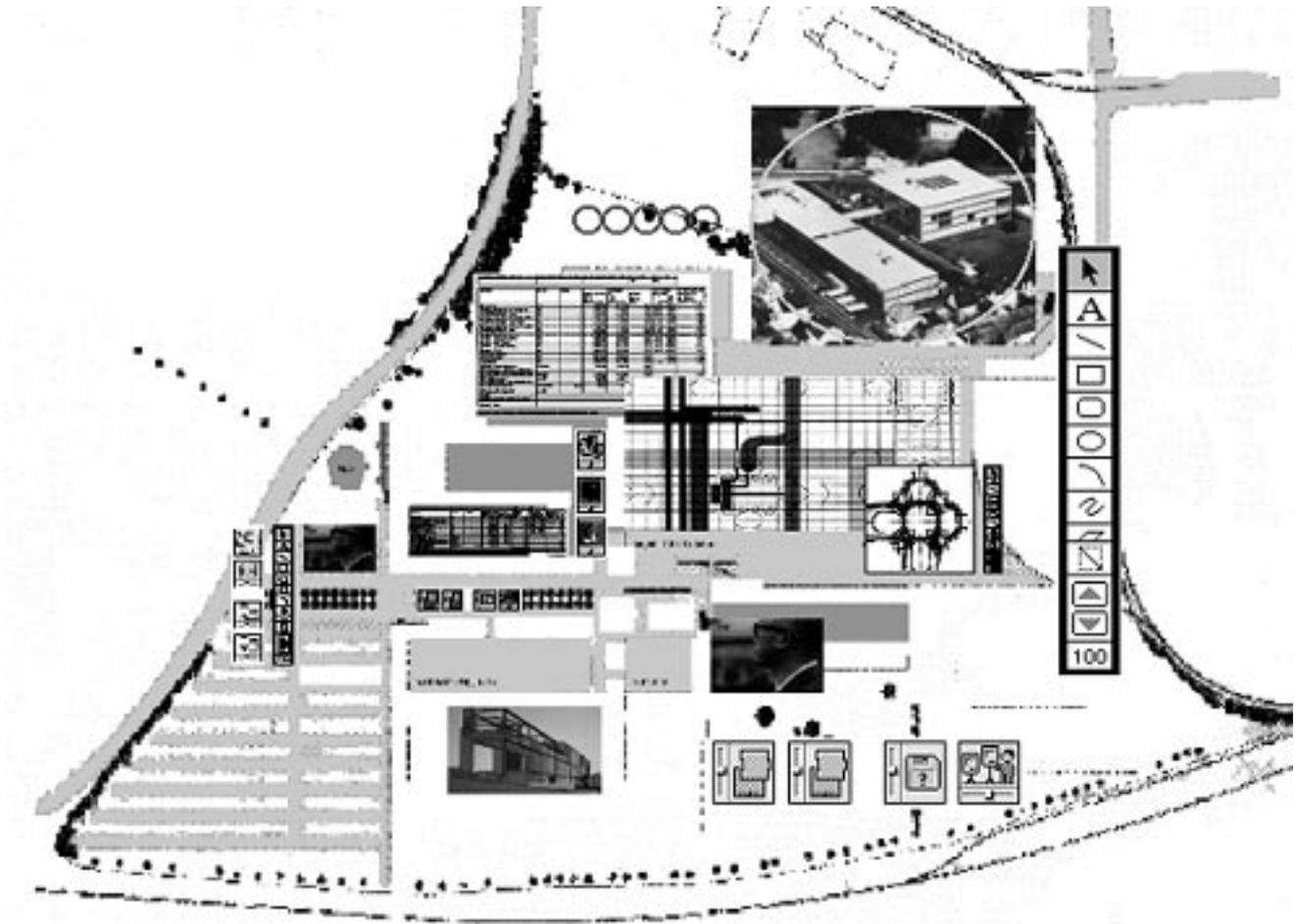


Abbildung 4: Beispiel für die Oberfläche

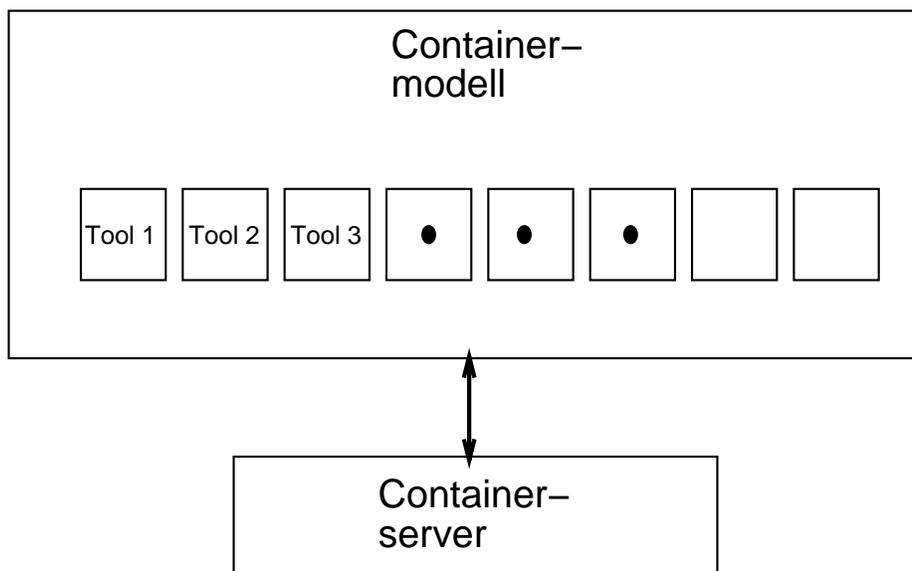


Abbildung 5: Prinzipieller Zusammenhang zwischen Containermodell und Server

- *Auflösung* beschreibt die verschiedenen Genauigkeitsstufen, in denen sich ein Objekt befindet. Beispielsweise kommt es bei einer Skizze nicht auf die exakten Maße an, sondern auf ungefähre Beziehungen zwischen einzelnen Objekten. In unserem Beispiel entspricht dies der Planung des Baukörpers und der Diskussion seiner äußeren Rahmenbedingungen. Bei der Detailplanung der Fassade sind dagegen die exakten Maße notwendig.
- Die *Größenordnung* beschreibt die verschiedenen Maßstäbe, in denen ein Gebäude geplant wird. Unser Beispiel aus Kapitel 2 ist beispielsweise im Maßstab 1:400 dargestellt. Dabei kann in jeder Größenordnung in allen Auflösungen geplant werden.
- Die Dimension *Alternative* gibt dem Planer die Möglichkeit, mehrerer Alternativen eines Entwurfsobjektes zu verwalten,
- *Teilsysteme* gibt den Typ des Objektes an und
- *Morphologie*, welchen Aspekt des Gebäudes (Nutzung, Erschließung, ...) das Entwurfsobjekt beschreibt.

Den kontinuierlichen Dimensionen sind immer konkrete Intervalle zugeordnet, während auf den diskreten Dimensionen durchaus auch Nullwerte, oder sog. “wild-cards” erlaubt sind. Dies beruht auf der Annahme, daß alle Entwurfsobjekte sowohl in der Geometrie als auch in der Zeit eindeutig beschrieben werden müssen, dagegen die restlichen Dimensionen keineswegs schon von Beginn an bekannt sind. Im ersteren Fall sprechen wir von *Zwangsdimensionen*, im zweiten Fall von *Freidimensionen*. Ist eine Dimension mit “wild-card” belegt, so entsteht anstatt eines Punktes im A4-Raum eine allgemeine Hyperfläche. Bei der Interpretation dieser Dimension werden dann alle potentiell möglichen Werte angenommen.

Die Interpretation der Werte der einzelnen Achsen ist von der Belegung der restlichen Achsen abhängig. So ist die Bedeutung der Bounding Box abhängig von der Belegung der Dimension Auflösung. Zusammen mit dem Zulassen von Nullwerten wird hierdurch ein wesentlicher Teil der Unbestimmtheit im architektonischen Entwurfsprozeß ausgedrückt.

Die Dimensionen bestimmen nun den Ausschnitt einer Entwurfentscheidung im Gesamtentwurf durch Festlegung der Position (Abbildung 6 oben). In unserem Beispiel werden alle Objekte wie die Laborräume, die Fassade, einzelne Teile der Fassade, ... durch die Beschreibung mit Hilfe von Positionen im Entwurf erfaßt. Abbildung 6 unten zeigt die Position des Sonnenschutzes auf der Südseite der Kantonsschule in Solothurn. Abbildung 7 zeigt die zugehörige graphische Repräsentation.

Der Referenzursprung des gesamten Gebäudes ist in Punkt 0 gelegt worden. Der Ortsvektor A für diese Beschreibung des Sonnenschutzes hat entsprechend die Koordinaten (8, 5, 0). Von dort ausgehend wird dann der Positionsvektor der rechten oberen Ecke der betrachteten Bounding Box angegeben. Der Sonnenschutz geht vom Erdboden bis an den Dachrand. Entsprechend ist der Richtungsvektor (74,2,9.6). Es handelt sich hierbei nicht um die exakten Maße für den Sonnenschutz, sondern nur um den ungefähren Ort, in dem er später angebracht werden soll. Der Grad der verwendeten Genauigkeit wird in der Dimension Auflösung verwaltet und in unserem konkreten Fall durch den Wert Hülle ausgedrückt. Die Bounding Box stellt bei dieser Auflösung einen noch ungenauen Bereich dar, in dem das Entwurfsobjekt lokalisiert werden muß. Wir erkennen des weiteren aus Abbildung 6, daß die Planung der Fassade dem Fachplaner Fassadenbauer obliegt, der gerade im Maßstab 1:400 den Sonnenschutz bearbeitet. Prinzipiell ist der Sonnenschutz zur Ausstattung eines Gebäudes zu zählen.

Die Art und Anzahl der Dimensionen zur Beschreibung eines Bereiches, ebenso wie deren Unterteilung in Zwangs- und Freidimensionen, sind keineswegs zwingend, sondern obliegen dem Ermessen des Planungsteams, das initial das System konfiguriert. An dieser Stelle ist eine große

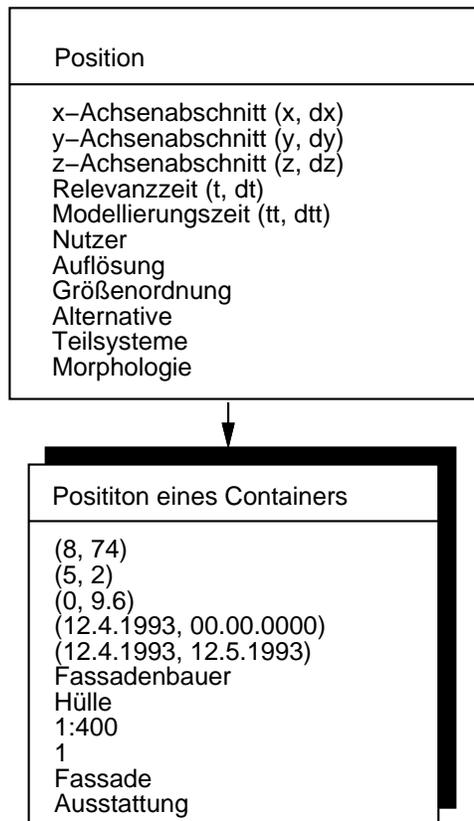


Abbildung 6: Beispiel für eine Position im A4-Raum

Offenheit des Konzeptes angestrebt, das es erlaubt, entsprechend den Anforderungen des jeweiligen Anwendungsgebietes die Dimensionen zu wählen. Auf diese Art und Weise kann das Konzept auf sehr unterschiedliche Anwendungsgebiete übertragen werden.

Zusätzlich zu den oben beschriebenen Dimensionen kann eine Planungsentscheidung durch eine ganze Reihe weiterer Eigenschaften charakterisiert werden. Dies sind zum einen Beschreibungsattribute, die

- die graphische Repräsentation des Containers auf der Oberfläche betreffen, wie z.B. 2D-Graphik, 3D-Graphik, Farbe, Transparenz etc.
- das durch die Graphik repräsentierte Bauteil betreffen, wie z.B. Kosten, Katalog-Nr., Gewicht etc..

Zum anderen sind es Funktionsattribute, die

- die Funktionalität des Containers auf der Oberfläche betreffen, wie z.B. Applikationen in Form von Textverarbeitung, Tabellenkalkulation etc.
- die Funktionalität des durch die Graphik repräsentierten Bauteils betreffen, wie die Steuerung einer Fassade.

Zusammenfassend spezifiziert sich ein Container also aus der Entwurfsentscheidung, seiner Position innerhalb des A4-Raumes, sowie seinen Beschreibungs- und Funktionsattributen.

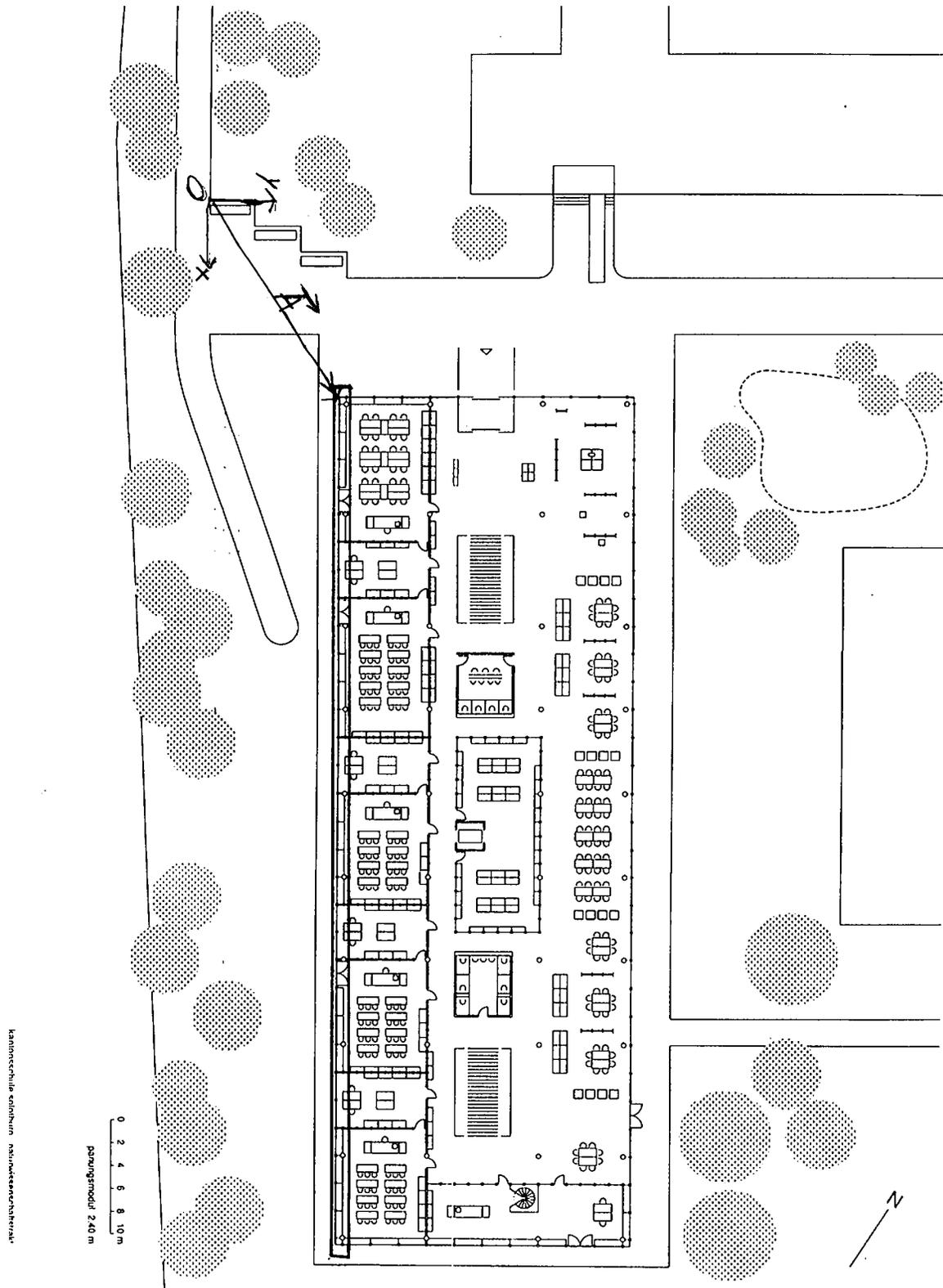


Abbildung 7: Planungsbeispiel: Bounding Box

### 3.4 Fachplanerintegration

Mit der zunehmenden Konkretisierung des Entwurfes, von der Entwurfsplanung hin zur Ausführungsplanung, werden zunehmend Fachaspekte integriert, die leichter formalisierbar sind als die stark intuitiven, kreativen Entscheidungen zu Beginn der Planung. Es handelt sich bei diesen Fachaspekten um Wissen von sog. Fachplanern, wie dem Statiker, dem Klimaingenieur oder dem Sanitärfachmann. Die Integration von Fachspekten geschieht durch eine unmittelbare Beteiligung der betreffenden Fachplaner in der Planung, also durch eine leistungsfähige Kommunikation, oder aber durch die Einbindung von Automaten, wie etwa Rechenprogrammen zur Kostenkalkulation oder zur Erstellung von Ausschreibungstexten.

Beispiele für rechnergestützte Fachplaner, die im Projekt ArchE Berücksichtigung finden, sind:

- **MIDI-Fachplaner:** Der Gebäudebaukasten MIDI besteht aus elementierten Bauteilen und einer Reihe von Anordnungsregeln. Diese beschreiben beispielsweise das Stützenraster, die Lage der Fassade, oder die maximale Spannweite der Träger
- **Armilla-Fachplaner:** Armilla ist ein allgemeines Installationsmodell, das die kollisionsfreie Koordinierung der Leitungen unterschiedlicher Medien zum Ziel hat. Die in Armilla definierten räumlichen Anordnungsregeln für Leitungen lassen sich sehr gut modellieren. Hinzu kommen gewerkspezifische Regeln wie etwa das minimale Gefälle einer Abwasserleitung, oder die maximale Länge eines Lüftungsrohres in seiner Abhängigkeit zum Rohrquerschnitt und dem geforderten Luftdurchsatz.

Alle Planungsentscheidungen der Fachplaner lassen sich ebenfalls als Container im A4-Raum ablegen. Sie werden gleichmäßig durch das Containermodell unterstützt, was eine Offenheit gegenüber dem Hinzufügen weiterer Fachplaner ermöglicht.

Geht man davon aus, daß auch die ersten Entwurfstätigkeiten des Architekten weitgehend rechnerunterstützt ablaufen, so bietet das Containermodell eine einheitliche Oberfläche, von der aus sich beliebige Fachplaner und Anwendungsprogramme aktivieren und Arbeiten zwischen den verschiedenen Komponenten koordinieren lassen. Aufgrund der eindeutigen Beschreibung der Planungsentscheidungen mit Hilfe der Positionsattribute ist ein koordiniertes Arbeiten auch auf verschiedenen Rechnern möglich. Die dabei notwendigen Koordinationsaufgaben werden vom Containerserver erbracht.

### 3.5 Abbildung auf Containerserver

Die Aufgabe des Containerservers ist die Bereitstellung der geforderten rechnergestützten Funktionalität des Containermodells. Die (technische) These unseres Vorhabens ist, daß dies besonders wirtschaftlich und elegant mit Hilfe einer objektorientierten Datenbank geschehen kann. Hierzu ist eine Abbildung des oben eingeführten Containermodells auf die Datenbank notwendig. In Abbildung 8 wird das hierzu von uns entwickelte konzeptuelle Schema dargestellt.

Die Container werden hier in einzelne Bestandteile aufgespalten. Diese werden jeweils als eigene Typen modelliert. Das Zentrum des Containers bilden seine Positionsattribute. Diese werden in dem Datentyp *Bereich* verwaltet.

Die Fachplanerintegration bedarf besonderer Vorkehrungen. Jedem Fachplaner unterliegt eine bestimmte Struktur, die alleine durch die Typen Beschreibungsobjekt, Funktionsobjekt und Bereich(sobjekt) nicht erfaßt werden können. Daher wird jedem Fachplaner die Möglichkeit eingeräumt, sein spezifisches Wissen in eigenen Objekthierarchien in der Datenbank zu verwalten. Diese Objekthierarchien bezeichnen wir als Partialmodelle.

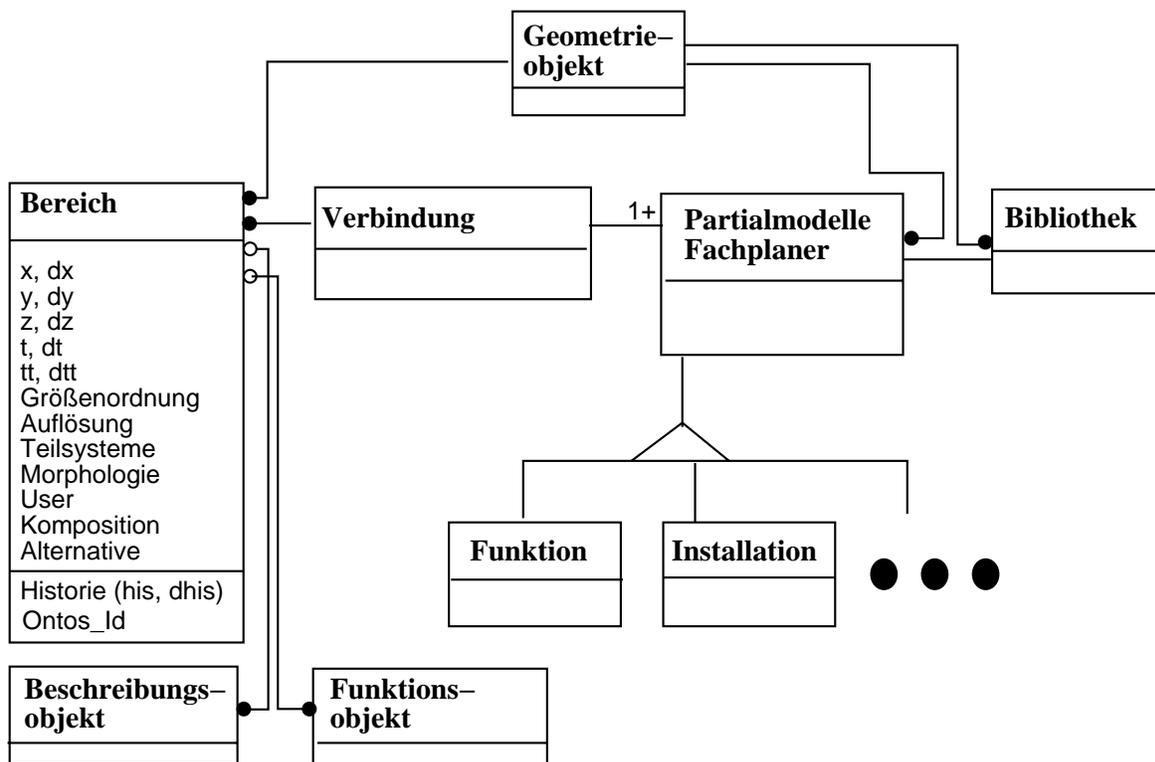


Abbildung 8: Konzeptuelles Schema in OMT-Darstellung

Die Beschreibungsattribute werden in zwei Teile aufgespalten: Ein Teil der Beschreibungsattribute ist für alle Entwurfsentscheidungen gleich, wie beispielsweise die Transparenz eines Bauteiles. Diese werden in einem separaten Objekt, dem Beschreibungsobjekt, verwaltet. Der andere Teil der Beschreibungsattribute wird über die Partialmodelle der einzelnen Fachplaner erfaßt. Die Funktionsattribute werden entsprechend behandelt. Zur Verwaltung konkreter Geometrien dient der Typ Geometrieobjekt.

Häufig werden auch schon vorgezeichnete Bauteile vom Containermodell angefordert. Die Bauteilbibliothek setzt die Anforderungen um.

Mit der Nichtlinearität des Entwurfsprozesses ist, wie zuvor erwähnt, gelegentlich auch ein Verwerfen von Entscheidungen und damit ein Rücksetzen auf frühere Entscheidungsstände verbunden. Gefordert wird also eine sehr feingranulare Rücksetzbarkeit von beliebigen Entwurfsausschnitten. Dies macht eine Rücksetzbarkeit auch auf Objektebene notwendig. Zur effizienteren Handhabung wird das Attribut Historie eingeführt, das wie eine Dimension des A4-Raumes behandelt wird. Demensprechend wird es in der Abbildung auf den Containerserver als Attribut des Types Bereich mit aufgenommen. History setzt sich analog zu den bisherigen Zeitattributen aus zwei verschiedenen Komponenten zusammen: *his* und *dhist*. Hierdurch wird das Intervall beschrieben, in dem ein Container samt seines Inhaltes unverändert blieb. Wird dieser verändert, so wird dies in der Datenbank wie ein neuer Container behandelt.

## 4 Anforderungen an das Constrainthandling

Wie bereits erwähnt, unterliegt der Entwurfsprozeß einer Vielzahl von Randbedingungen. Da deren Einhaltung in die Zuständigkeit des unterstützenden Rechensystems - und das heißt hier: Containerserver - fallen soll, müssen diese Bedingungen zunächst im Containermodell ausdrückbar sein. Dazu genügen die in Abschnitt 3.1 eingeführten Attribute nicht. Es muß also eine

zusätzliche Möglichkeit der Formulierung der Bedingungen existieren. Diese Formulierung werden wir unter dem Begriff Constraint fassen, während wir ihre Abbildung im Containerserver als Konsistenzbedingung bezeichnen wollen.

Constraints stellen einen integralen Bestandteil des architektonischen Entwurfes dar und sind selbst wieder Entwurfsentscheidungen. In [1, 5] werden die Anforderungen an die Unterstützung von Constraints aufgezeigt. Im ersten Teil dieses Kapitels werden wir die Eigenschaften von Constraints analysieren. Diese definieren dann die von dem Containermodell geforderte Funktionalität zur Konsistenzsicherung, die der Containerserver zur Verfügung zu stellen hat.

Im zweiten Teil des Kapitels wird dann auf die Handhabung der Constraints im Containermodell eingegangen. Hierbei beschreibt ebenfalls die *Position* (analog zu physikalischen Entwurfsentscheidungen) den Teil eines Entwurfes, in dem ein bestimmtes Constraint erfüllt sein soll. Sie ist ebenfalls durch die Positionsattribute des A4-Raumes beschrieben. Constraints lassen sich damit zu Entwurfsausschnitten in Bezug setzen.

Zum Abschluß dieses Kapitels wird dann auf die Abbildung der Constraints auf den Containerserver eingegangen.

## 4.1 Eigenschaften von Constraints

Die in diesem Abschnitt vorgestellten Eigenschaften bzw. Anforderungen sind direkt mit den am Projekt beteiligten Architekten in vielen Diskussionen erarbeitet worden. Sie legen die vom Containermodell geforderte Funktionalität zur Konsistenzsicherung fest.

**Zeitlich und räumlich partielle Gültigkeit von Constraints:** Vielschichtigkeit und Nicht-linearität bedingen, wie wir gesehen haben, ein häufiges Wechseln zwischen den betrachteten Entwurfsausschnitten. In jedem dieser Entwurfsausschnitte werden ständig neue Constraints, die in der Regel nur einen Teil des gerade betrachteten Ausschnittes betreffen, festgelegt, indem den Constraints eine Position zugeordnet wird. Die Zuordnung zwischen Position und Constraints ist nicht fest, sondern sogar hochgradig dynamisch. Insbesondere kann diese Zuordnung dynamisch eingerichtet und auch wieder gelöst werden. Dies liegt an der Arbeitsweise des Architekten: Je nachdem, wo und was ein Architekt momentan plant, kommen unterschiedliche Mengen von Constraints hinzu. Die Position eines Constraints ist hierbei nicht nur geometrischer Natur, sondern umfaßt analog zu Entwurfsobjekten auch Charakterisierungen wie die momentan betrachtete Detaillierungsstufe oder den Fachplaner, für den dieses Constraint gelten soll. Typisches Beispiel hierfür ist das Einfügen des neuen Laborraumes, der eine Menge von Constraints an den Ausschnitt stellt, in dem er geplant werden soll, wobei nicht von vornherein klar ist, wo der Raum plaziert wird. Da der zusätzliche Laborraum für jede Art von Fachunterricht geeignet sein soll, müssen hier wesentlich mehr Constraints erfüllt werden als bei den restlichen Laborräumen, die von vornherein für einen bestimmten Fachunterricht vorgesehen sind.

**Gleiche Constraints mit unterschiedlichen Positionen:** Ein Constraint kann durchaus mehreren Positionen zugeordnet sein. Sollen etwa zwei Laborräume mit denselben Anforderungen entstehen, dann gelten die hierzu zugeordneten Constraints in den beiden separaten Räumen, die der Architekt für diese Laborräume vorsieht. Für einen dritten Laborraum kann jedoch durchaus ein separater Katalog von Constraints vorgesehen werden.

Betrachten wir zu diesen Überlegungen den in Abbildung 9 dargestellten Entwurfsausschnitt. Dort sind einige Constraints angegeben. Jeder Lehrer verlangt in dem von ihm zu bestimmenden Entwurfsausschnitt die spezifische Anzahl von Waschbecken. Zwei der Laborräume sollen mit 5 Waschbecken ausgestattet sein, ein dritter mit nur 3 Waschbecken. Hierbei ist einem der Lehrer gleichgültig, ob sich diese Waschbecken in seinem Lehrmittelraum oder im Klassenraum

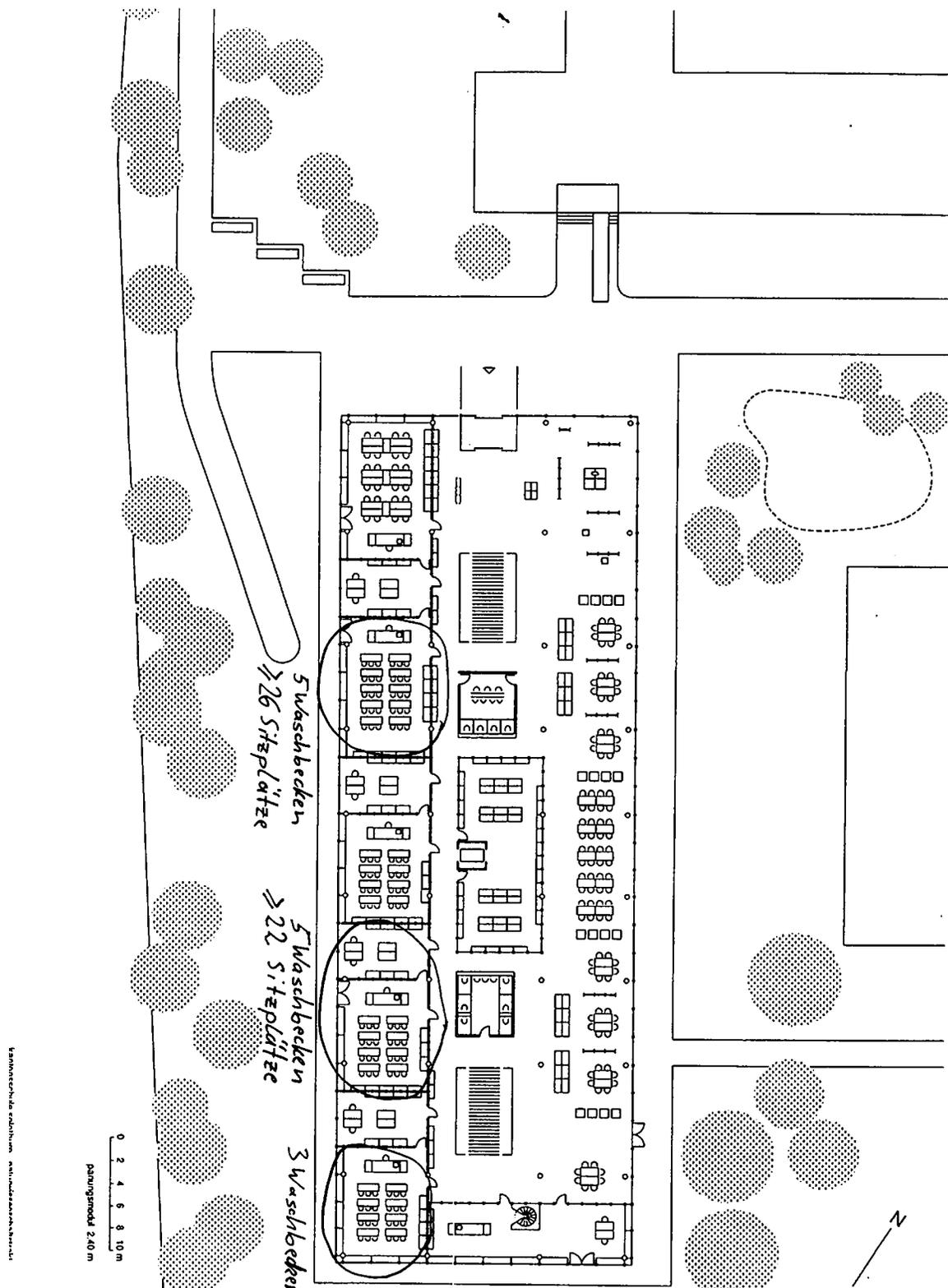


Abbildung 9: Planungsbeispiel: Constraints

befinden (mittlere Position in Abbildung 9). Die Positionen der Constraints sind durch die Ovale markiert. Ein Architekt selektiert die von ihm gewünschten Positionen und ordnet ihnen dann die entsprechende Constraints zu.

**Steuerbare Überprüfungszeitpunkte für Constraints:** Bei der Festlegung einer Position liegt die Annahme einer langfristig geplanten Gültigkeit von Constraints in dem durch die Position beschriebenen Planungsausschnitt zugrunde. Es ist aber durchaus denkbar, daß das Constraint zunächst noch gar nicht erfüllbar ist oder auch gar nicht überprüft werden soll. Der Zeitpunkt einer gewünschten Überprüfung muß daher durch den Architekten einzeln festlegbar sein. Dies kann dadurch geschehen, daß die Überprüfung von Constraints ereignisabhängig erfolgt. Will man etwa überprüfen, ob die Laborräume die an sie gestellten Anforderungen an die Leitungsversorgung erfüllen, kann dies frühestens zu einem Zeitpunkt geschehen, zu dem die Lage der Fachräume grob festgelegt ist. Eine weitere ereignisabhängige Überprüfung ist, daß nach dem direkten Einfügen bestimmter Bauteile, wie der Sonnenschutz auf einer Seite der Fassade, abgeprüft werden muß, ob diese Seite auch wirklich nach Süden ausgerichtet ist oder ob hier ein besonderer Sonnenschutz unnötig ist. Die Forderung nach ereignisabhängiger Überprüfung hat ihren Ursprung in verschiedenen Charakteristika:

- *Zustandsänderung:* Beim Entwurf variieren die zu betrachtenden Constraints zu den unterschiedlichen Planungszuständen. Jeder von diesen Zuständen hat seine eigenen Constraints, die mit überwacht werden müssen. Prüfzeitpunkte für solche Zustände treten insbesondere in Zusammenhang mit den in der HOAI festgelegten Meilensteinen auf. Es werden dementsprechend aufgrund des Erkennens eines Meilensteines eine ganze Reihe von Constraints überprüft.
- *Explizite Aktivierung:* Aufgrund eines expliziten Ereignisses, wie das Einfügen einer Stütze, wird die Überprüfung bestimmter Constraints notwendig. Das Einfügen des Laborraumes muß die Überprüfung der Constraints, die ein Laborraum erfüllen muß, auslösen.
- *Umfeldaktivierung:* In einem bestimmten Bereich wird eine Aktion, z.B. das Einfügen eines neuen Laborraumes, ausgeführt. Dies führt nicht nur zu der Überwachung der Constraints, die explizit zu dem Laborraum gehören, sondern es müssen alle Constraints betrachtet werden, die durch die Änderung im entsprechenden Entwurfsausschnitt betroffen sein könnten. Ein solches Ereignis könnte etwa das zusätzliche Einziehen eines Deckenträgers im darunterliegenden Raum sein. Diese Änderung pflanzt sich noch weiter fort: Es muß in diesem Bereich die Leitungsführung in der Decke neu überprüft werden, da zusätzliche Hindernisse hinzugekommen sind.

**Unterschiedliche Reaktion auf Verletzungen und deren bewußte Tolerierung:** Es muß die Möglichkeit geben, auf Verletzungen von Constraints flexibel zu reagieren. In einem Fall muß die Verletzung sofort behoben werden, wie eine geplante Abwasserleitung, die nicht genügend Gefälle aufweist<sup>7</sup>. Im anderen Fall muß die Verletzung nicht sofort behoben werden, da die zugehörigen Constraints eher vage sind, wie die Anforderung, daß der Filmraum neben dem neuen Laborraum liegen sollte. Hier ist dem Architekten - lediglich als eine Art Erinnerung - zu melden, daß er momentan dieses Constraint verletzt; dies hat aber keinerlei zwingende Folgen auf den Planungsablauf. Wie streng ein Constraint jeweils eingehalten werden sollte, ist von dem jeweiligen Bauherren bzw. Architekten abhängig. An dieser Stelle muß ein System,

---

<sup>7</sup>Tatsächlich ist die Abwasserleitung die kritischste Leitung in einem Gebäude, da hier sehr harte Randbedingungen gelten und die Abwasserleitung nicht, wie fast alle anderen Leitungen, mit einer Pumpe versehen werden kann.

das Akzeptanz beim Benutzer erreichen will, sehr flexibel sein. Diese Forderung ist insbesondere dadurch begründet, daß Inkonsistenzen zwischen einzelnen Entwurfsentscheidungen eher die Regel als die Ausnahme darstellen.

**Unterschiedliche Reaktion auf Verletzung gleicher Constraints in unterschiedlichen Positionen:** In den unterschiedlichen Positionen können auf dasselbe verletzte Constraint unterschiedliche Reaktionen folgen. In unserem Beispiel kann jeder Lehrer die Anforderungen an seinen Laborraum selber bestimmen. Hierbei kommt es durchaus vor, daß dieselben Constraints an verschiedene Laborräume gestellt werden. Allerdings haben die Lehrer den Constraints unterschiedliche Wichtigkeiten zugeordnet, so daß Verletzungen dieser Constraints verschiedene Reaktionen erfordern. Einem Lehrer ist es beispielsweise besonders wichtig, daß er fünf Wasserbecken in seinem Raum hat (siehe Abbildung 9). Entsprechend muß, falls diese Anforderung durch die bereits existierenden Wasserleitungen nicht erfüllt werden kann, eine tiefgreifende Änderung im Leitungssystem vorgenommen werden. Der andere Lehrer hat dieselbe Anforderung gestellt. Ihm ist es aber auch wichtig, einen zusätzlichen Dunstabzugsarbeitsplatz zu bekommen. Ist die von ihm geforderte Anzahl von 5 Wasserbecken nur auf Kosten des zusätzlichen Dunstabzugsarbeitsplatzes möglich, so verzichtet er lieber auf ein Wasserbecken.

**Modifikation der Reaktion auf Constraintverletzungen:** Fast zwangsläufig folgt, daß die Reaktion auf die Verletzung ein- und desselben Constraint auch änderbar sein muß. Wird mit einem Lehrer beispielsweise über einen unerfüllbaren Constraint an seinem Raum gesprochen, so kann es sein, daß dieser aufgrund der Argumente der Architekten die Prioritäten der Constraints und damit auch die notwendige Reaktion auf Verletzung von Constraints ändert. Solche Änderungen sind typisch für den architektonischen Entwurfsprozeß: Constraints ändern häufig im Laufe eines Entwurfes ihre Bedeutung und damit die erwünschte Reaktion bei ihrer Verletzung.

**Zustandsanzeige für beliebige Entwurfsausschnitte:** Aufgrund der Forderung, inkonsistente Zustände zu tolerieren, wird es auch notwendig, in jedem beliebigen Entwurfsausschnitt die Übersicht über die dort erfüllten bzw. nicht erfüllten Constraints zu wahren. Ein Architekt hat beispielsweise gerade den neuen Laborraum in das Gebäude eingefügt und will nun an einem anderen Entwurfsausschnitt weiterarbeiten. In diesem neuen Ausschnitt ist es für ihn aber notwendig zu wissen, welche Constraints noch nicht erfüllt sind, damit er einen Überblick über die momentan aktuelle Entwurfssituation in diesem Entwurfsausschnitt und Anhaltspunkte für die nächsten Aufgaben erhält.

**Zuschalten von Constraints:** Mit der zusätzlichen Forderung nach einem Filmraum fallen eine Vielzahl neuer Constraints an, die nun in das Gesamtsystem eingebracht werden müssen. Da diese Constraints zu Entwurfsbeginn nicht vorhersagbar waren, kann keine ereignisgesteuerte Aktivierung stattfinden. Die Constraints müssen explizit neu in den zu überwachenden Katalog mit aufgenommen werden.

**Vordefinieren von Constraints:** Ein Architekt will nicht bei jedem Einbringen neuer Constraints in den Entwurf diese neu formulieren, sondern er will Constraints, die er schon einmal formuliert hat, wieder benutzen können. Das selbe gilt für Ereignisse und Aktionen. Hierzu sollten schon formulierte Constraints, Ereignisse und Aktionen in Bibliotheken verwaltet werden, die dann sukzessive erweitert werden können.

**Fachplanerconstraints:** Constraints stellen einen wesentlichen Teil der Beschreibung der Fachplaner dar, wie z.B. die Neigungen, die eine Abwasserleitung mindestens haben muß, oder der maximale Abstand eines Wasseranschlusses zu der ihn versorgenden Stammleitung. Fachplaner werden im Laufe des Entwurfes nach und nach mit in den Entwurf einbezogen. Hierbei müssen dann alle Constraints eines hinzutretenden Fachplaners auf einmal in den Entwurf eingebracht werden.

Auch für die Beschreibung der Fachplanerconstraints sind wesentlich die Ereignisse, die eine Überprüfung bestimmter Constraints auslösen und die Aktionen, die auf die Verletzung von Constraints folgen sollen. Diese Zusammenhänge zwischen Constraints, Ereignissen und Aktionen müssen schon bei der Formulierung des Fachplaners definierbar sein, da sie wesentliches Wissen der Fachplaner darstellen. Wird beispielsweise eine Stammleitung für Warmwasser verlegt, so ist es nicht ausreichend, alle Leitungen entsprechend nachzuführen, sondern es muß zusätzlich noch die Bedingung überprüft werden, ob nun keine der Zuleitungen zu lange ist. Bei Kaltwasserleitungen gibt es keine derartige Bedingung. Sie müssen dementsprechend auch nicht überprüft werden, falls die Stammleitung versetzt wird.

Constraints gelten häufig nicht für den gesamten Entwurf, sondern beispielsweise lediglich in bestimmten Größenordnungen oder bestimmten Auflösungen. Das Wissen, in welchem solchen Umfeld Constraints gelten, ist ebenfalls vorab bestimmbar.

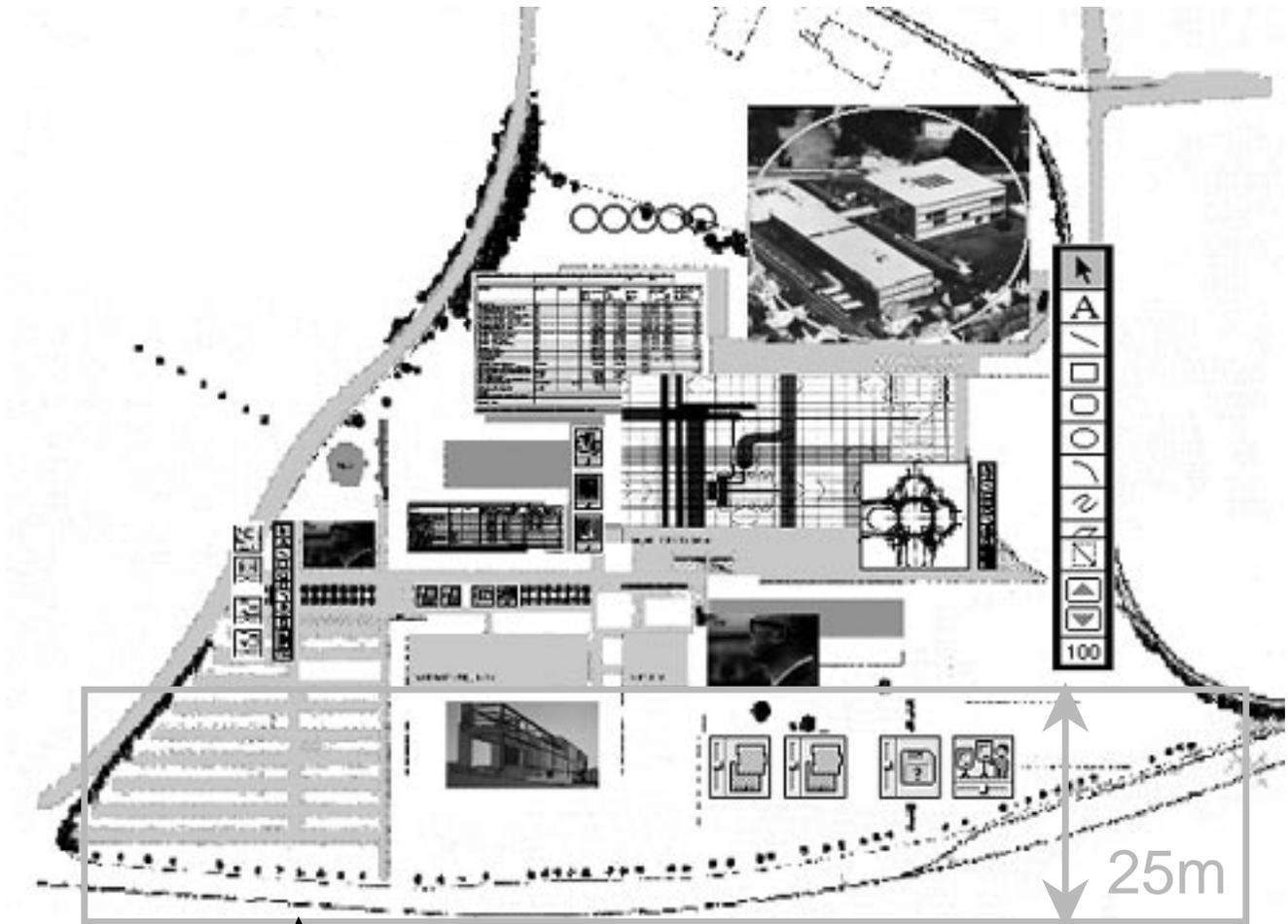
**Aktivierung der Fachplanerconstraints:** Der architektonischer Entwurf ist hochgradig dynamisch. Es kann hier nicht davon ausgegangen werden, daß diese Fachplaner auf einmal für den gesamten Entwurf aktiviert werden, sondern es muß möglich sein, Fachplaner in bestimmten, vom Anwender ausgewählten Bereichen beliebig zuschalten zu können. Dementsprechend dürfen die oben beschriebenen Kombinationen von Constraints, Ereignissen und Aktionen nur in diesem Bereich aktiviert und somit der entsprechenden Position zugeordnet werden.

**Rücksetzbarkeit in beliebigen Entwurfsausschnitten:** Ein Entwurf wird sehr häufig wieder zurückgesetzt. Dieses Rücksetzen geschieht in von den Benutzern definierten Entwurfsausschnitten. Durch dieses Rücksetzen sind die Constraints, deren Positionen sich mit diesem Ausschnitt überlappen, ebenfalls betroffen: Es muß ausschließlich in dem rückzusetzenden Ausschnitt der Zustand bezüglich der Constraints wieder hergestellt werden, der zu dem Rücksetzzeitpunkt gültig war. Es muß hier auch berücksichtigt werden, daß nun der zurückgesetzte Entwurfsausschnitt nicht notwendigerweise mit seinem Umfeld konsistent ist. Entsprechende Prüfungen müssen veranlaßt werden.

## 4.2 Benutzung von Constraints

Constraints und Konsistenzprüfung sollten sich harmonisch in die in Abbildung 5 dargestellte Systemarchitektur einpassen. Da Constraints im Verständnis der Architekten Entwurfsentscheidungen repräsentieren, ist es nur natürlich, wenn Constraints genauso wie physikalische Entwurfsentscheidungen durch Container dargestellt werden.

Abbildung 10 zeigt, wie Architekten Constraints auf der graphischen Oberfläche benutzen wollen. Augenblicklich betrachte der Architekt den städtebaulichen Aspekt der Planung. Er lege fest, daß ein gewisses Gebiet ein Wohngebiet sein soll. Mit dem Treffen dieser Entwurfsentscheidung fallen eine Reihe von Constraints an, die nun erfüllt sein sollten. In unserem Beispiel lauten die Bedingungen, daß die Richtlinien "Lärmschutz Wohngebiet" und "min. Abstand Fabrik = 25 m" eingehalten werden müssen. Diese Constraints sollen in dem mit Wohngebiet beschrifteten Ausschnitt gelten. Er ist demnach die Position der Constraints. Nicht explizite Angaben für die Belegung von Positionsattributen werden hierbei aus dem aktuellen Entwurfsausschnitt automatisch entnommen.



Wohngebiet

constraint 3  
 prioritat 1

Bedingung :  
 Laermschutz Wohngebiet  
 min. Abstand Fabrik 25 m

Abbildung 10: Beispiel für Constraints

Zusätzlich zu der Formulierung des Constraints werden noch weitere Eigenschaften angegeben. Darunter fallen für den Architekten unter anderem die Bedingung selbst, die Ereignisse, die eine Überprüfung auslösen, und die Aktionen, die im Falle einer Verletzung ausgeführt werden. In unserem Beispiel soll das Constraint beispielsweise die “Priorität 1” haben, dies bezieht sich auf die Art der Behandlung des Constraints: Diese Constraints müssen immer erfüllt sein.

Die vorangegangenen Überlegungen führen dann dazu, daß Constraints nicht mehr, wie in traditionellen Ansätzen üblich, bestimmten Objekten oder Objekttypen zugeordnet sind, sondern einer Position. Wir wollen dies im folgenden anhand eines schematisierten Entwurfs illustrieren. In Abbildung 11 ist dazu der Zusammenhang zwischen Constraints, den Positionen, in denen sie gelten, und dem gerade durch einen Architekten bearbeiteten Ausschnitt aus einem Entwurf schematisch dargestellt. Die Abbildung beschreibt die folgende Situation:

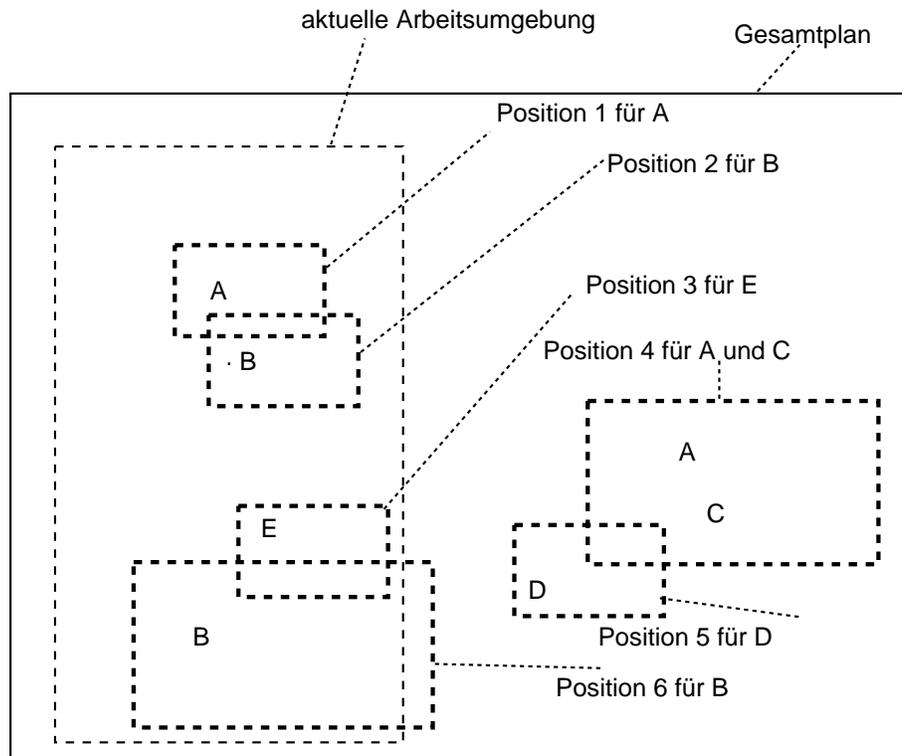


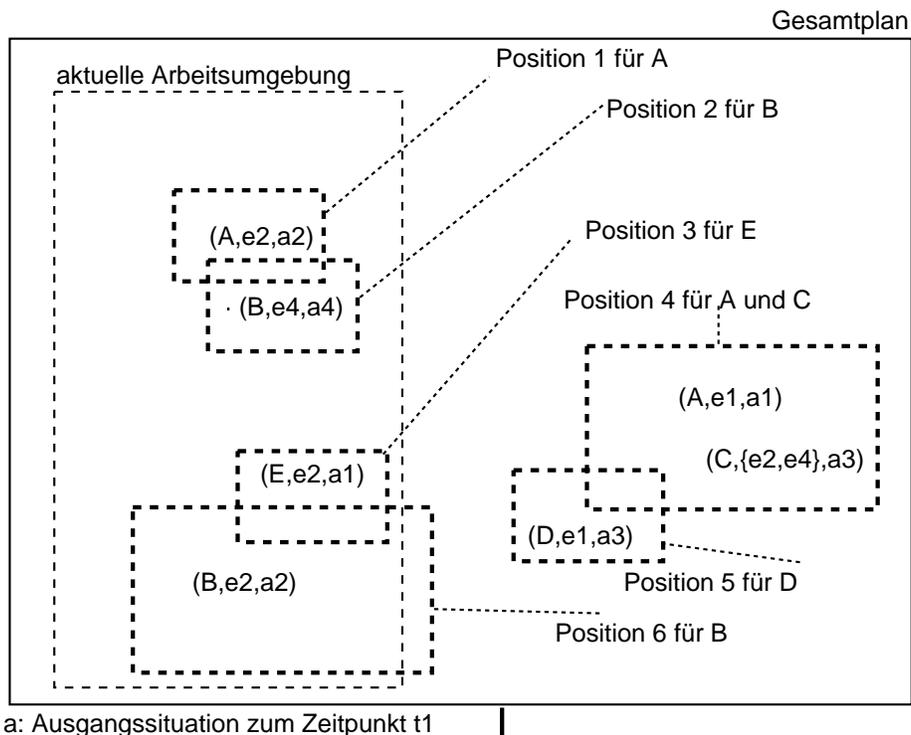
Abbildung 11: Zusammenhang Positionen und Constraints

Der *Gesamtplan* repräsentiert den gesamten Entwurf. Innerhalb dieses Gesamtplans arbeitet ein Architekt oder ein Fachplaner auf einem bestimmten Ausschnitt, der in Abbildung 11 *aktuelle Arbeitsumgebung* genannt wird. *A*, *B*, *C*, *D* und *E* sind Constraints. Die Rechtecke, in denen sie notiert sind, stellen die Positionen dar, in denen sie definiert sind.

Ein Constraint (etwa *A* oder *B*) kann nun in verschiedenen Positionen gültig sein. Ebenso können in derselben Position (etwa Position 4) verschiedene Constraints gelten. Die Positionen der verschiedenen Constraints können sich beliebig überlappen. Genauso kann sich die aktuelle Arbeitsumgebung mit beliebig vielen Positionen von Constraints überschneiden.

Zusätzlich sind im vorangegangenen Abschnitt noch weitere Eigenschaften von Constraints eingeführt worden, wie eine Überprüfung auslösende Ereignisse und im Falle der Verletzung auszuführende Aktionen. Abbildung 12 zeigt die erforderlichen Ergänzungen.

In jeder zugeordneten Position kann die Überprüfung eines Constraint durch unterschiedliche Ereignisse ausgelöst werden, und die Verletzung des Constraints kann unterschiedliche Reaktionen hervorrufen. In Abbildung 12 wird beispielsweise in Position 1 die Überprüfung von



↓  
Änderung zum Zeitpunkt t2

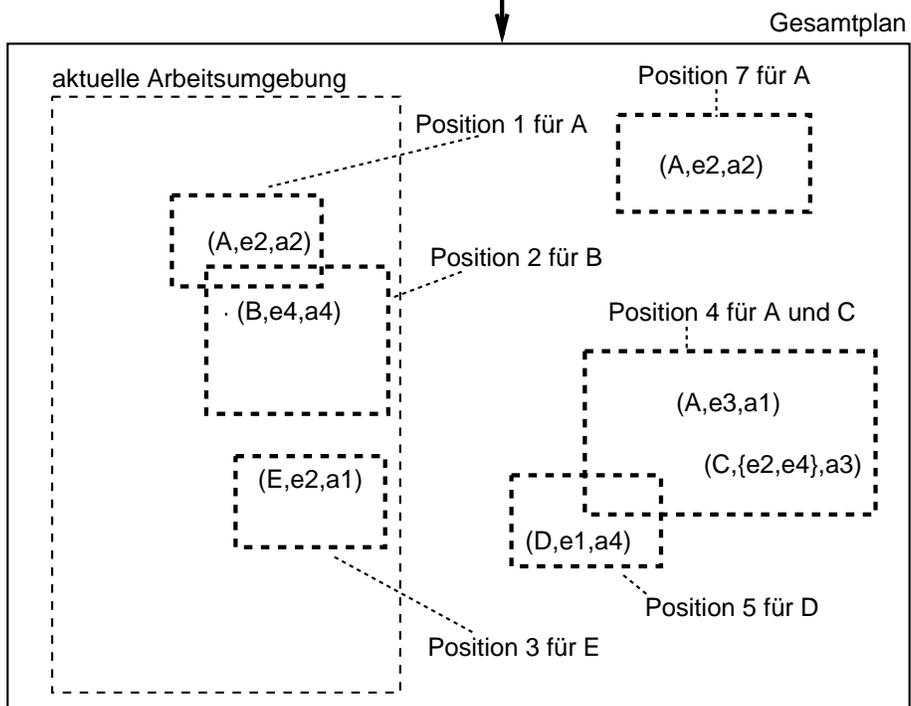


Abbildung 12: Positionen, Constraints, Ereignisse und Aktionen

Constraint A durch das Ereignis e2 ausgelöst, und es wird, falls in dieser Position A verletzt ist, die Aktion a2 ausgeführt. Während dessen wird in Position 4 dasselbe Constraint A durch das Vorkommen des Ereignisses e1 überprüft und im Falle der Verletzung die Aktion a1 verursacht.

Für jeden auf der Oberfläche identifizierbaren Ausschnitt ergibt sich eine andere Menge von zu betrachtenden Constraints: Alle Constraints, deren Positionen sich mit dem momentan betrachteten Entwurfsausschnitt überlappen, müssen bei einer Überprüfung berücksichtigt werden. In unserem Beispiel wären das, falls gerade die aktuelle Arbeitsumgebung betrachtet wird, die Tripel (A,e2,a2), (B,e4,a4), (E,e2,a1) und (B,e2,a2). Die Constraints A, B, E müssen überprüft werden, falls eines der ihnen zugeordneten Ereignisse eingetreten ist. Bemerkenswert ist hierbei, daß eine Überprüfung des Constraints B in unterschiedlichen Positionen, die sich beide mit der aktuellen Arbeitsumgebung schneiden, durch unterschiedliche Ereignisse ausgelöst wird. Entsprechend können, je nach dem ob eine Verletzung in Position 2 oder in Position 6 geschieht, unterschiedliche Aktionen ausgelöst werden.

Ein Entwurfsprozeß ist hochgradig dynamisch. Ständig werden schon getroffene Entscheidungen revidiert, alte Entscheidungen werden wieder aufgegriffen oder neue Ideen werden ausprobiert. Dieses Vorgehen spiegelt sich direkt in den für den Entwurf relevanten Constraints und den Positionen, in denen sie gelten, wieder. Wie bereits ausgeführt, werden diese ebenfalls im Laufe eines Entwurfes, immer bezogen auf bestimmte Ausschnitte, verändert, zurückgesetzt oder wieder aufgegriffen. Abbildung 12 stellt mögliche Änderungen schematisiert dar. Es finden hier zu einem Zeitpunkt t2 eine Reihe von Änderungen statt: Das Constraint in Position 6 soll beispielsweise in Zukunft nicht mehr gelten, und die zu Constraint B gehörende Position 2 soll vergrößert werden. Constraints werden in neuen Positionen aktiviert, wie dies für das Constraint A in der Position 7 der Fall ist. Desweiteren können Änderungen auch die zugeordneten Ereignisse oder Aktionen betreffen. Dies geschieht in Position 4 für das Constraint A und in Position 5 für das Constraint D.

Als reversible und modifizierbare Entwurfsentscheidungen sind also Constraints in der hier betrachteten Entwurfsumgebung eine recht fluide Angelegenheit. Aus Nutzersicht hat aber jedes Constraint bei aller Fluidität einen festen Bezugspunkt: eine Position. Dies motiviert unsere wesentliche Idee, der Position eine zentrale Rolle bei der Handhabung von Constraints in einem so hochdynamischen Umfeld zuzuordnen. Es entsteht das in Abbildung 13 dargestellte Abhängigkeitsverhältnis. Constraints, Ereignisse und Aktionen sind separat zu sehen, die Zuordnung geschieht über die zugehörige Position. Diese Vorgehensweise paßt sich nahtlos und damit völlig natürlich in die in Abschnitt 3.1 eingeführte Arbeitshypothese der Entwurfsräume ein.

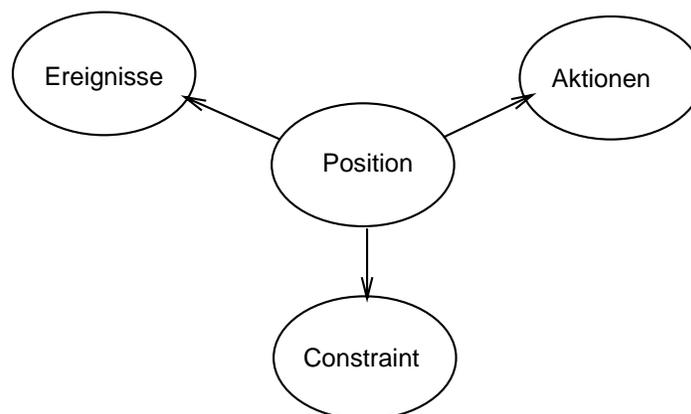


Abbildung 13: Position im Zentrum

### 4.3 Einbringen von Constraints

Ähnlich wie in Kapitel 3 müssen auch für das Einbringen von Constraints zwei Facetten betrachtet werden: Die Aktionen auf der Benutzungsoberfläche und deren Auswirkungen im Containermodell als dem Datenmodell der Anwendungen. Dieser Abschnitt zeigt den generellen Umgang des Entwerfers mit den Constraints im Containermodell. Prinzipiell kann hier zwischen Anfragen an den Zustand des Entwurfes bezüglich der Constraints und dem Ändern des Zustandes bezüglich der Constraints unterschieden werden:

- **Zustandsanzeige:** Der Entwerfer läßt sich die in einem bestimmten, von ihm ausgewählten Ausschnitt geltenden Constraints anzeigen. Zusätzlich benötigt er dann noch Informationen darüber, welche Constraints in diesem Abschnitt momentan nicht erfüllt sind, um Anhaltspunkte zu erhalten, wo er als nächstes seine Arbeit fortsetzen soll.
- Sollen neue Constraints hinzugefügt oder überhaupt erst welche eingeführt werden, so markiert der Benutzer zuerst den Ausschnitt aus dem Entwurf, für den das zu definierende Constraint relevant ist. Anschließend hat er zwei Möglichkeiten:
  - Über eine möglichst einfache Constraintsprache werden neue Constraints formuliert und in die Datenbank eingebracht.
  - Das Constraint, das der Benutzer einbringen will, ist bereits in einer Bibliothek in der Datenbank vorhanden. Der Benutzer kann es selektieren und in dem markierten Bereich aktivieren.

Entsprechend zu der Formulierung der Constraints können weitere Eigenschaften festgelegt werden, wie Ereignisse, die eine Überprüfung auslösen sollen oder die Reaktionen, die eine Verletzung nach sich zieht. Hierbei existieren für die Ereignisse und Aktionen ebenfalls Bibliotheken, auf die der Benutzer, entsprechend den Constraints, im Bedarfsfall zurückgreifen kann. Der Ausschnitt aus dem Entwurf entspricht der in Kapitel 3 eingeführten Position. Alle Dimensionen erhalten hier prinzipiell vorerst eine Defaultbelegung, die sich aus den Umgebungsvariablen des im Containermodell markierten Entwurfsausschnittes zusammensetzen:

- Die geometrischen Attribute werden vom gerade aktuellen Entwurfsausschnitt abgeleitet.
- Die Zeitattribute bekommen ebenfalls die entsprechenden Werte, die aus der Systemzeit abgeleitet werden.
- Die diskreten Attribute übernehmen ihre Werte defaultmäßig aus der zuletzt vom selben Fachplaner benutzten Position.

Anschließend hat der Entwerfer die Möglichkeit, diese Werte seinen Wünschen entsprechend abzuändern.

Im Datenmodell werden die Constraints im Sinne einer homogenen Lösung über **Constraint-container** erfaßt, die die Gesamtheit aus Position, zugeordnetem Constraint und allen weiteren beschreibenden Eigenschaften beinhalten. Abbildung 14 zeigt nochmals, auf welche Bestandteile zur Bildung von Constraintcontainern zurückgegriffen wird.

Abbildung 15 zeigt analog zu Abbildung 6 auf, welche Komponenten ein Architekt beispielsweise für die Beschreibung eines Constraints benötigt. An der Südseite der Kantonsschule in Solothurn wird bei der Fassade ein Sonnenschutz benötigt. Da dieses Constraint direkt die Fassade der Südseite betrifft, ist die Position, die der Architekt für dieses Constraint angibt, gleich der Position aus Abbildung 6. Zusätzlich zu Position und Constraint selbst gibt der Architekt weitere Eigenschaften an: Er legt fest, daß das Constraint immer überprüft wird, falls ein Teil

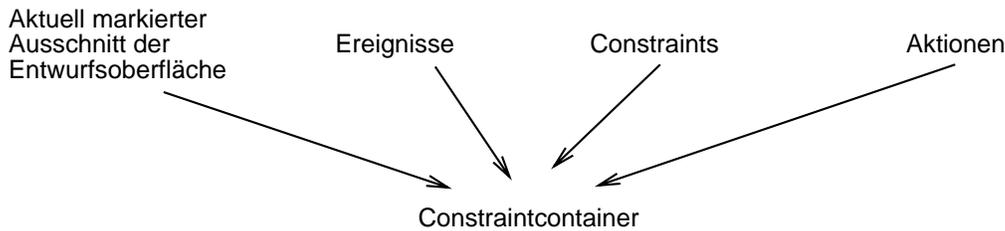


Abbildung 14: Die Bestandteile eines Constraintcontainers

der Fassade innerhalb der Position verändert wird, und daß im Falle einer Verletzung neben ihm noch der Klimafachplaner benachrichtigt werden soll.

Ist der Constraintcontainer einer konkreten Position des Entwurfes zugeordnet, so wird von **aktiven Constraintcontainer** gesprochen. Derartige Container können, wie bisher beschrieben, unmittelbar auf der Benutzungsoberfläche konstruiert werden. Andererseits existieren jedoch mit den Fachplanern vorabdefinierte Constraints samt ihrer beschreibenden Eigenschaften. Deren Formulierung geschieht unabhängig vom konkreten Entwurf. Hierbei braucht der Architekt die Möglichkeit, zusammengehörige Constraints, Ereignisse und Aktionen festzulegen. Zusätzlich kann er durch die explizite Belegung der Positionsattribute zusätzliches Wissen über zusammengehörige Constraints, Ereignisse und Aktionen einbringen, beispielsweise in welcher Größenordnung diese von Interesse sind. Abbildung 16 demonstriert wie ein Architekt einen derartigen **vorabdefinierten Constraintcontainer** festlegt. Hierzu kann er wiederum auf die in Bibliotheken abgelegten Ereignisse, Bedingungen und Aktionen zurück greifen, oder aber er formuliert sich die einzelnen Bestandteile selbst. Zusätzlich kann er noch bestimmte Positionsattribute spezifizieren, die dann später als Selektionskriterien benutzt werden können. So können alle Constraints, die dem Teilsystem Kaltwasser zugordnet sind, mit einer Anfrage selektiert werden.

Abbildung 17 zeigt einen Constraintcontainer, der zur allgemeinen Beschreibung des Fachplaners Fassadenbauer gehört. Dieser war die Basis für den Constraintcontainer in Abbildung 15. Bei der Definition dieser Regel gibt der Benutzer die Ereignisse, das Constraint und die Aktionen an. Anschließend kann er noch im Bereich der Positionsattribute weitere Spezifikationen angeben. In unserem Beispiel gibt er noch an, daß dieses Constraint zum Fachplaner Fassadenbauer gehört, daß es die Fassade direkt betrifft und daß es lediglich zur Betrachtung der Ausstattung von Interesse ist. Die restlichen Positionsattribute sind in diesem Fall mit "wild-card" belegt. Aufgrund der Spezifikation des Nutzers "Fassadenbauer" kann dann dieses Constraint als zum Fachplaner Fassadenbauer identifiziert werden.

Wird nun ein Fachplaner in einem bestimmten Ausschnitt des Entwurfes aktiviert, so muß der vorabdefinierte Constraintcontainer in einem aktiven Constraintcontainer überführt werden. Dazu muß die bisher unvollständige Position mit der aktuell auf der Oberfläche markierten Position in Verbindung gebracht werden. Dies geschieht derart, daß diejenigen Positionsattribute, die in der vorabdefinierten Position ein "wild-card" enthalten, die aktuellen Werte zugewiesen bekommen, während die restlichen ihren vorabdefinierten Wert behalten.

Aufgrund der hohen Dynamik des architektonischen Entwurfes werden bestehende Constraintcontainer sehr häufig verändert. Hierzu läßt sich der Entwerfer den ursprünglichen Constraintcontainer anzeigen. Diesen kann er dann beliebig modifizieren.

#### 4.4 Überprüfung

Bisher wurde beschrieben, wie Constraintcontainer in den Entwurf eingebracht werden, nicht aber wann sie überprüft werden sollen. Hierbei werden vom Benutzer verschiedene Fälle unter-

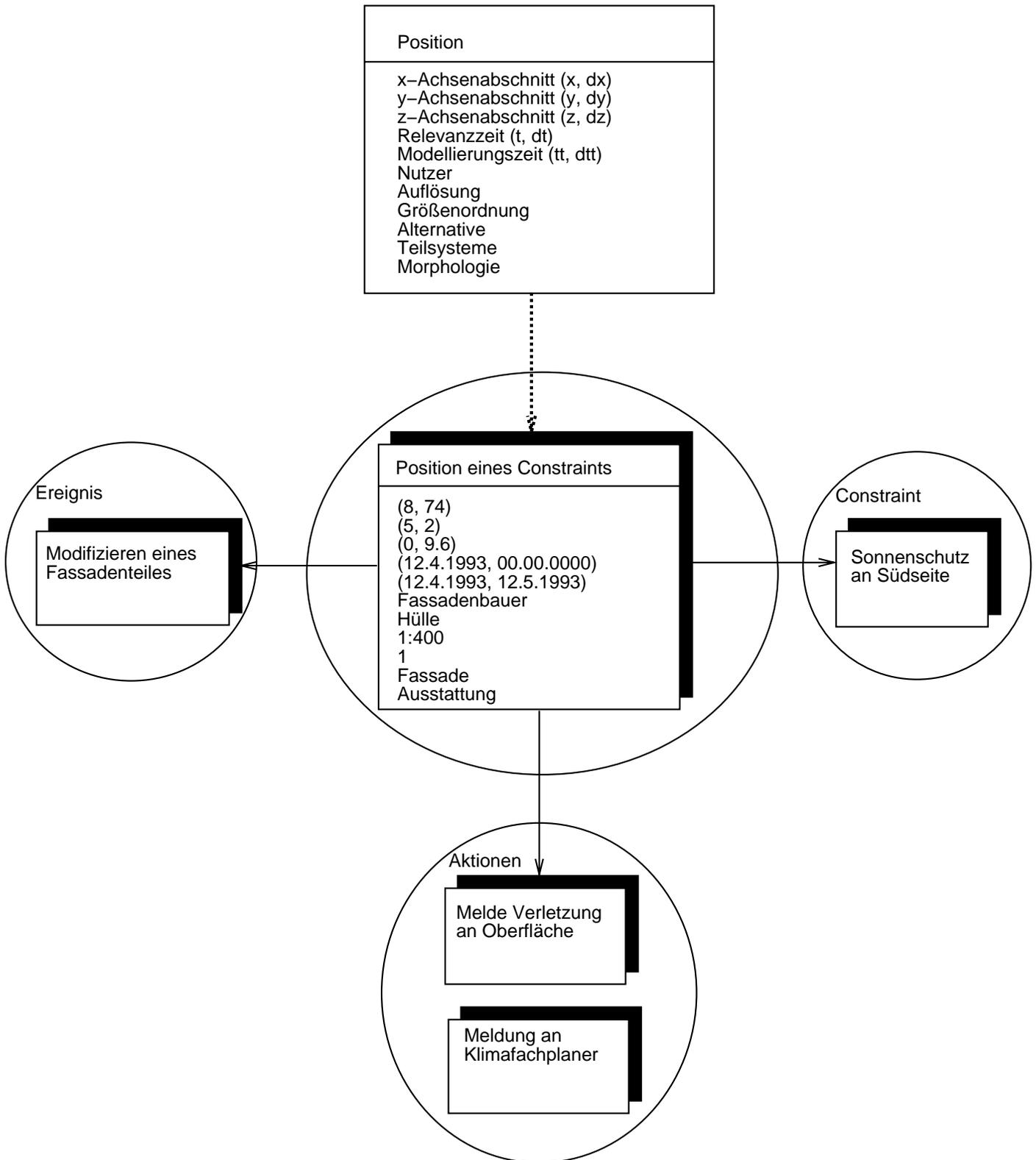


Abbildung 15: Beispiel eines Constraintcontainers

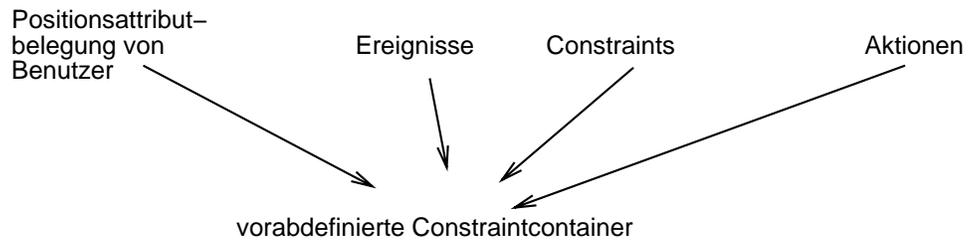


Abbildung 16: Vorab definierter Constraintcontainer

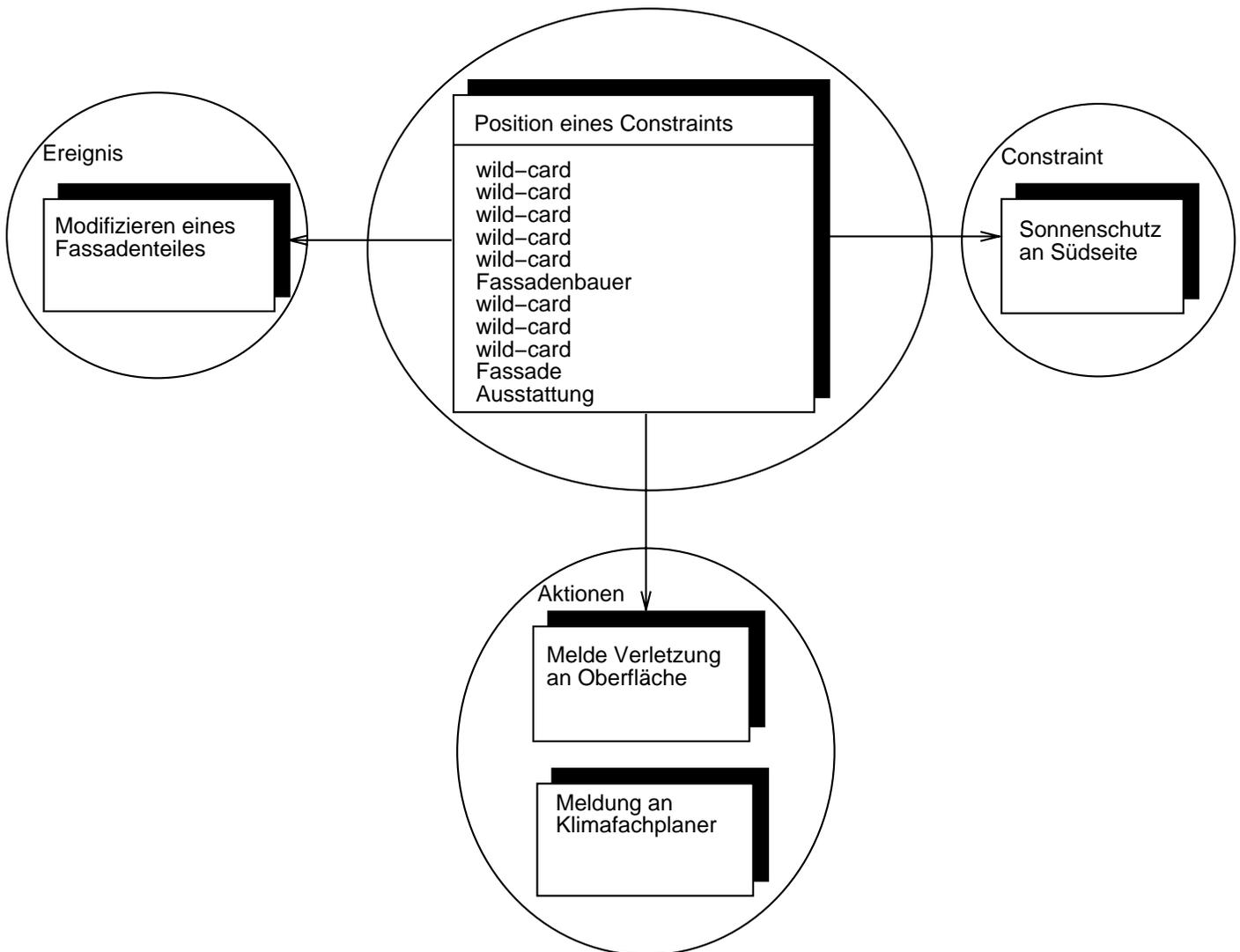


Abbildung 17: Beispiel eines Fachplanerconstraintcontainers

scheiden:

- Ein Constraint soll überprüft werden, falls diejenigen Ereignisse, die ein Benutzer angegeben hat, eingetreten sind.
- Ein Constraint soll prinzipiell immer überprüft werden.

Der Eintritt eines Ereignisses wird hierbei ebenfalls positionsgebunden betrachtet. D.h. ein Ereignis hat eine Auswirkung auf einen Constraintcontainer, falls seine Position innerhalb dessen Position fällt. Daher müssen alle Constraintcontainer überprüft werden, auf deren Position ein bestimmtes Ereignis eine Auswirkung zeigt.

Abbildung 18 zeigt diese Situation auf. E1, E2 und E3 sind Ereignisse, die vom Benutzer in seiner aktuellen Arbeitsumgebung verursacht werden. CC1, CC2 und CC3 repräsentieren Constraintcontainer. CC1 liegt vollständig innerhalb der aktuellen Arbeitsumgebung, während CC2 sich mit ihr lediglich überlappt. CC3 befindet sich vollständig außerhalb der aktuellen Arbeitsumgebung. Da der Benutzer lediglich Ereignisse innerhalb seiner aktuellen Arbeitsumgebung auslösen kann, müssen überhaupt nur CC1 und CC2 prinzipiell überprüft werden. Da nun aber E1 völlig außerhalb von CC1 und CC2 liegt, müssen bei Anstoß einer Überprüfung aufgrund von E1 weder CC1 noch CC2 überprüft werden. E2 liegt innerhalb von CC2, d.h. die Auswirkung, die E2 im Entwurf hervorruft, überlappt sich mit der Position von CC2. In diesem Fall muß CC2 mit überprüft werden, nicht jedoch CC1. Entsprechendes gilt für E3 und CC1.

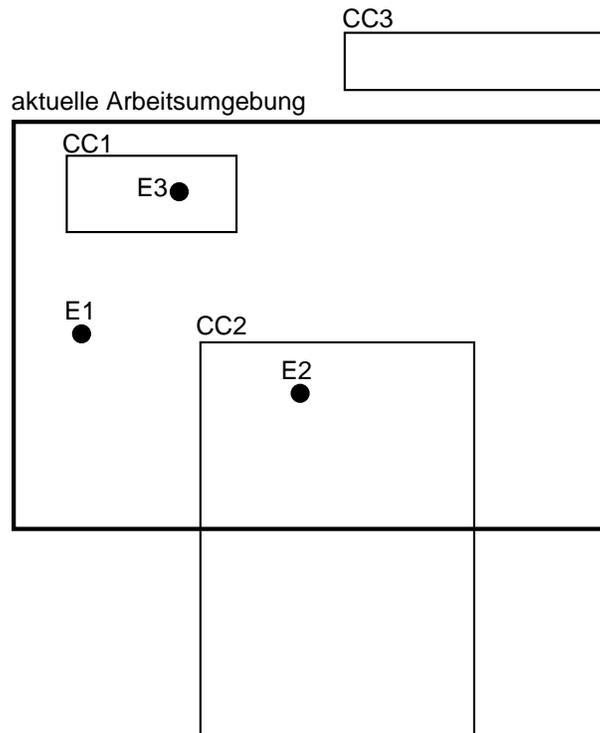


Abbildung 18: Zusammenhang Ereignisse und Constraintcontainer

#### 4.5 Abbildung auf den Containerserver

Der oben vorgestellte Constraintcontainer wird wie alle Container im Containerserver verwaltet. Hierzu ist analog zu Kapitel 3.5 eine konzeptuelle Modellierung der Constraintcontainer

notwendig. Abbildung 19 zeigt die notwendigen Erweiterungen zu der in Abbildung 8 dargestellten Modellierung der Container auf.

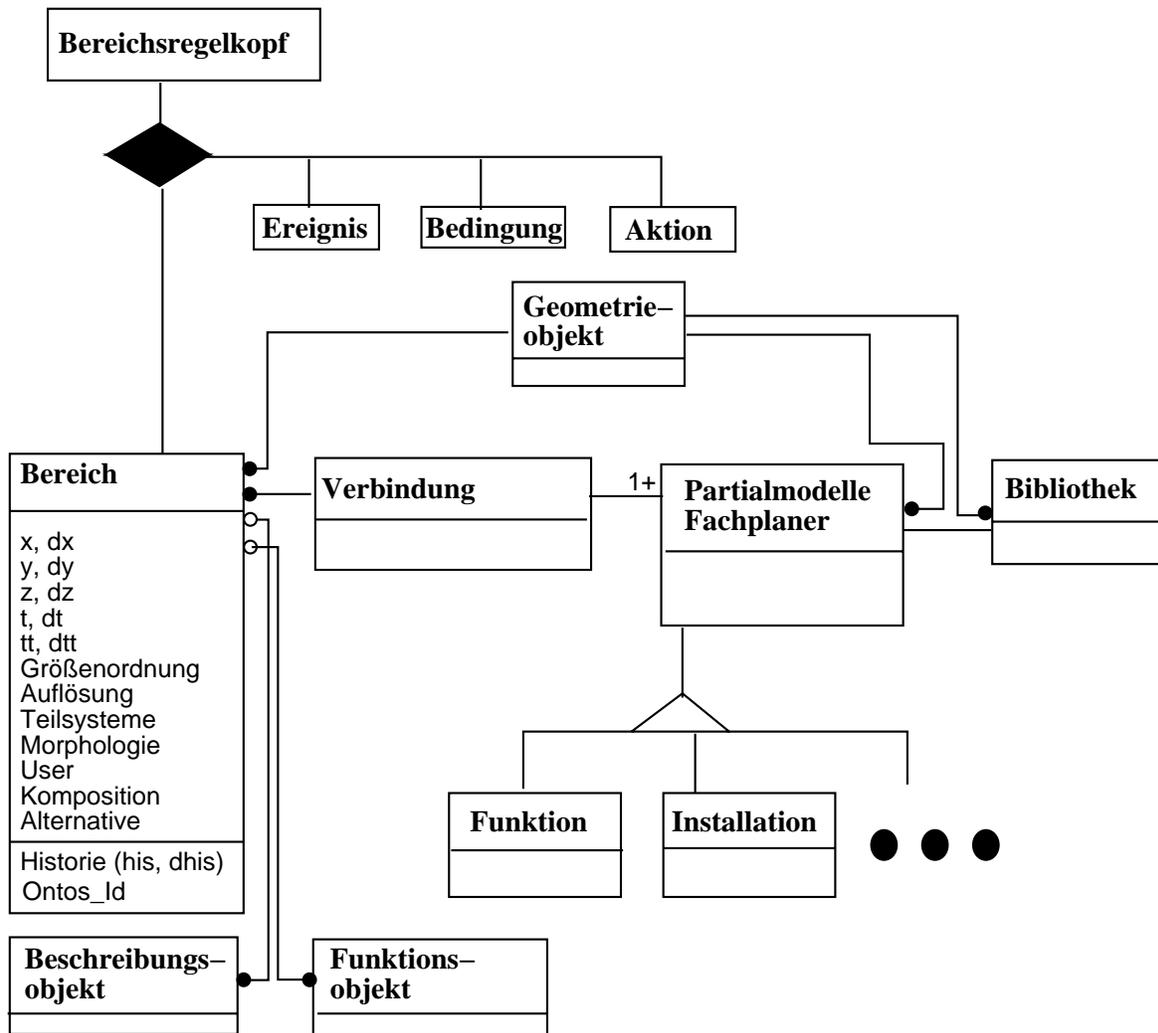


Abbildung 19: Erweitertes konzeptuelles Schema

Die Position eines Constraints wird dementsprechend wieder auf den Typ Bereich abgebildet. Die durch die Constraints festgelegten Bedingungen werden in einer Typhierarchie, die (hier nicht weiter ausgeführt) durch den Objekttyp “(Konsistenz)Bedingung” erfaßt wird, repräsentiert. Die eine Überprüfung auslösenden Ereignisse und die im Falle der Verletzung auszuführenden Aktionen werden ebenfalls in eigenen Typhierarchien verwaltet. Die Zusammengehörigkeit der einzelnen Teile, wie er in Abbildung 15 dargestellt ist, wird über den Bereichsregelkopf erfaßt (Abbildung 20).

## 5 Constraintüberwachung durch den Containerserver

In diesem Kapitel wird nun ein Konzept vorgestellt, mit dem die oben beschriebenen Anforderungen an einen Mechanismus zur Handhabung der Constraints durch den Containerserver erfüllt werden können. Insbesondere muß die Struktur aus Abbildung 13 durch den Containerserver unterstützt werden.

Vom Architekten werden Constraints und physikalische Entwurfsentscheidungen im A4-Raum analog behandelt. Daraus folgt vor allem, daß wiederum Bereiche als Abbildung von Positionen eine zentrale Rolle spielen.

## 5.1 Bereichsregeln

Die konzeptuelle Modellierung, die zum Einbringen der Constraints in die Datenbank benötigt wird, wurde im vorangegangenen Kapitel dargestellt. Abbildung 20 zeigt nochmal den Ausschnitt aus der Modellierung, der die Constraints betrifft. Zusammengehörige Bereichsregelköpfe, Bereiche, Ereignisse, Konsistenzbedingung und Aktionen werden nun als eine Bereichsregel bezeichnet.

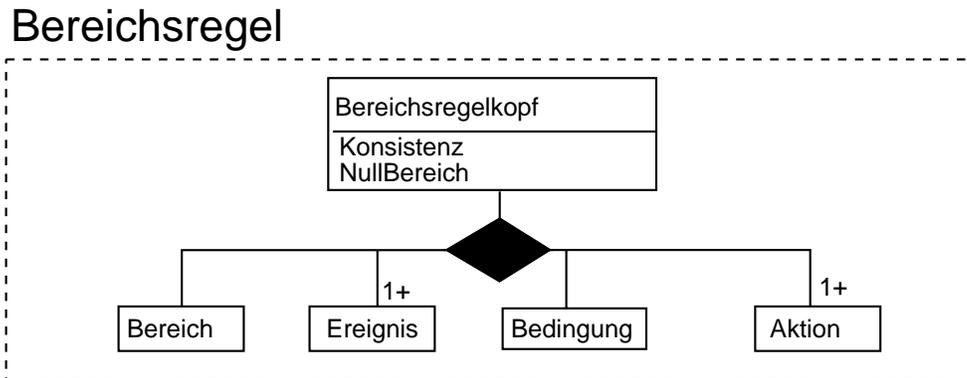


Abbildung 20: Modellierung der Abhängigkeiten

Abbildung 21 zeigt zwei Bereichsregeln, die aus dem schematisierten Beispiel in Abbildung 12 resultieren. Für die Position 4 sind 2 verschiedene Constraints, A und C, aktiviert, die durch die Bedingungen A und C beschrieben werden. Entsprechend sind zur Formulierung dieses Sachverhaltes die beiden Regeln R1 und R2 notwendig. Beide referenzieren die Position 4, die in den Bereich4 abgebildet wird. R1 drückt aus, daß die Konsistenzbedingung A in Bereich4 überprüft wird, falls Ereignis e1 eintritt. Im Falle der Verletzung wird dann die Aktion a1 ausgeführt. R2 beschreibt, daß die Bedingung C überprüft wird, falls in Bereich4 die Ereignisse e2 oder e4 eingetreten sind. Im Falle der Verletzung wird hier a3 aktiviert.

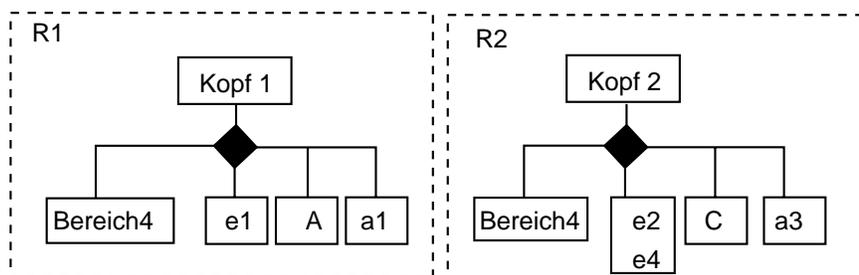


Abbildung 21: Beispiele für Bereichsregeln

Abbildung 22 zeigt ein Beispiel einer Bereichsregel, das die Abbildung des Constraintcontainers aus Abbildung 15 im Containerserver darstellt.

Gemäß Abbildung 13 bilden Constraint, Ereignisse und Aktionen bezüglich einer Position eine Einheit. Diese Einheit wird im Containerserver durch den Bereichsregelkopf verkörpert. Die Einführung eines eigenständigen Bereichsregelkopfes bietet darüber hinaus Vorteile:

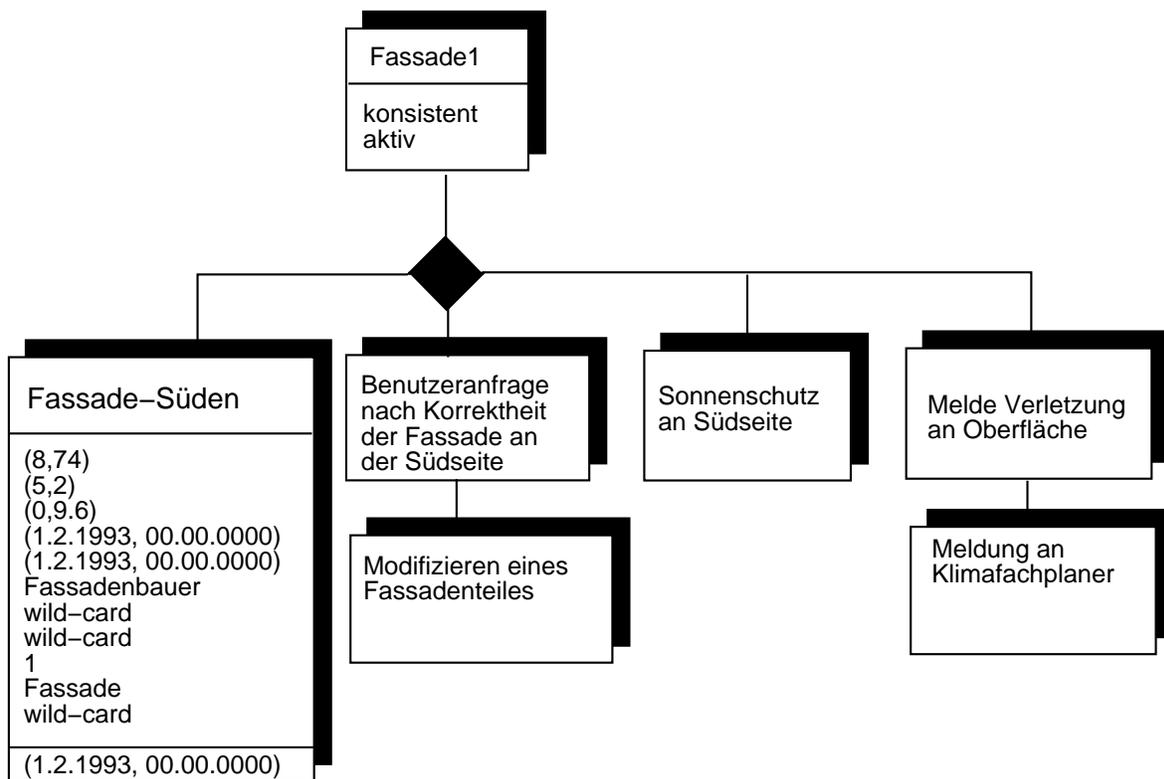


Abbildung 22: Beispiel einer Bereichsregel

- Einiges an Semantik der Anwendung kann so über Kardinalitäten in der Modellierung erfaßt werden: Abbildung 20 stellt dar, daß sich ein Bereichsregelkopf aus genau einem Bereich, mindestens einem Ereignis, genau einer Konsistenzbedingung und mindestens einer Aktion aggregiert:
  - **Genau ein Bereich und eine Bedingung pro Bereichsregelkopf:** Dies ist nach dem in Abschnitt 4.2 und dem eingangs Gesagten selbstverständlich. Die Position ist Bezugspunkt, daher kann nur genau ein Bereich existieren. Ein Constraint ist die Einheit der Überprüfung, daher muß eine Bedingung vorhanden sein, und es darf nur eine einzige sein. Dies schließt nicht aus, daß der selbe Constraint in mehreren verschiedenen Positionen aktiviert sein kann, und daß in jeder dieser Positionen die Überprüfung dieses Constraints durch unterschiedliche Ereignisse ausgelöst werden kann und im Falle der Verletzung unterschiedliche Aktionen angestoßen werden.
  - **Mehrere Ereignisse und Aktionen pro Bereichsregelkopf:** Eine Überprüfung eines Constraints kann in derselben Position durch mehrere unterschiedliche Ereignisse ausgelöst werden, und mehrere Aktionen können im Falle einer Verletzung ausgeführt werden.
- Es lassen sich dem Bereichsregelkopf Attribute zuordnen, die es erlauben, globale Informationen zu verwalten. Bisher sind zwei Attribute, *Konsistenz* und *NullBereich* eingeführt.
  - Das Attribut *Konsistenz* verwaltet den aktuellen Zustand der Konsistenzbedingung im zugeordneten Bereich. Hier kann vermerkt werden, ob eine Bedingung in diesem Bereich gilt, nicht erfüllt oder noch nicht überprüft ist. Diese Zustandsanzeige wird benötigt, um in jedem beliebigen Ausschnitt des Entwurfes Aussagen über den aktu-

ellen Zustand der Konsistenz machen zu können, um z.B. eine erneuerte Überprüfung zu starten.

- Das Attribut *NullBereich* ist für die Fachplanerintegration notwendig und wird später erläutert.
- Mit dem Bereichsregelkopf läßt sich auch eine sehr feingranulare Mitführung der Historie der Constraints erreichen: Wird einer der beteiligten Bestandteile verändert, so wird eine neue Ausprägung einer Bereichsregel generiert, indem ein neuer Bereichsregelkopf angelegt wird. Die unveränderten Teile werden als solche übernommen, und der veränderte Teil wird entsprechend eingefügt. In den betroffenen Bereichen müssen dann die Historie-Attribute angepaßt werden.

## 5.2 Vorabdefinition

Die Organisation von Bereichsregeln in fünf getrennte Bausteine gestattet es, Ereignisse, Bedingungen und Aktionen jeweils vorab zu definieren und ohne eine konkrete Zuordnung zu einem Bereichsregelkopf mit in die Datenbank aufzunehmen. Dies eröffnet die Möglichkeit, Bibliotheken von Ereignissen, Bedingungen und Aktionen aufzubauen. Auf diese Bibliotheken kann ein Benutzer dann bei der Neuformulierung dieser Teile jeweils zurückgreifen.

## 5.3 Zustandsübergänge der Bereichsregeln

In Kapitel 4 sind zwei verschiedene Arten von Constraintcontainer angesprochen worden: Zum einen aktivierte Constraintcontainer, die aktuell mit überwacht werden, und zum anderen vorabdefinierte Constraintcontainer, die zur Unterstützung der Fachplanerintegration notwendig sind. Dies erfordert eine entsprechend unterschiedliche Abbildung im Containerserver: A-Regeln (aktive Bereichsregeln) und N-Regeln (nicht aktive Bereichsregeln). Beim Zuschalten eines Fachplaners können N-Regeln auf einfache Art und Weise während des Entwurfes, auch paketweise, in A-Regeln umgewandelt werden, indem lediglich der Bereich der N-Regel durch den neuen Bereich der A-Regel ersetzt wird. Dazu wird zunächst die N-Regel kopiert. Anschließend wird in dieser Kopie der Bereich ausgetauscht. Die Ereignisse, die Bedingung und die Aktionen werden unverändert übernommen, und die Attribute des Bereichsregelkopfes werden entsprechend angepaßt. Auf diese Art und Weise können dieselben N-Regeln in vielen verschiedenen Bereichen aktiviert werden.

Zur Unterscheidung der A- und N-Regeln wird im Bereichsregelkopf das Attribut "Null-Bereich" eingeführt. Im Falle einer Überprüfung müssen lediglich die aktiven Regeln geprüft werden.

Alle vorab definierten Constraintcontainer, die den selben Fachplaner beschreiben, sind im Bereichsattribut "user" als zu diesem Fachplaner gehörend markiert. Dies bietet eine einfache Möglichkeit, alle Constraintcontainer eines Fachplaners mit einer einfachen Anfrage zu aktivieren.

## 5.4 Dynamik der Bereichsregeln

In diesem Abschnitt soll anhand des Beispiels aus Abbildung 12 dargestellt werden, wie sich Änderungen im Containermodell auf die Bereichsregeln im Containerserver auswirken. Insbesondere muß durch die Datenhaltung der geforderte feingranulare Rücksetzmechanismus ermöglicht werden. Deshalb spielt das Attribut Historie eine zentrale Rolle: Dieses Attribut zeigt an, in welchem Zeitintervall eine bestimmte Bereichsregel unverändert war. Ändert ein Constraint und damit eine Konsistenzbedingung ihre Rolle im Entwurf, so wird jedesmal eine neue Instanz der betroffenen Bereichsregel nach dem folgenden Vorgehen generiert:

1. Kopiere Ausgangsbereichsregel  $R_a$  und erhalte  $R_b$ .
2. Der Endpunkt des Historie-Intervalles von  $R_a$  wird auf die aktuelle Systemzeit gesetzt.
3.  $R_b$  bekommt die aktuelle Systemzeit als Anfangszeitpunkt zugewiesen. Als Endpunkt wird eine Zeit weit in der Zukunft eingetragen.
4. In  $R_b$  werden nun die durch den Benutzer geforderten Veränderung an der Regel durchgeführt.

Abbildung 12 zeigt den Zustand bezüglich der aktiven Bereichsregeln zu zwei verschiedenen Zeitpunkten an, wobei die Situation aus Abbildung 12a zum Zeitpunkt  $t_2$  in die Situation Abbildung 12b abgeändert wird. Die jeweils zugehörigen Bereichsregeln finden sich in Abbildung 23, wobei den Positionen der Constraints Bereiche der zugeordneten Konsistenzbedingungen entsprechen.

Wie oben aufgeführt werden für eine Konsistenzbedingung, wenn sie in unterschiedlichen Bereichen gilt, verschiedene Ausprägungen von Bereichsregeln angelegt. Die Konsistenzbedingung A gilt zum Zeitpunkt  $t_1$  in den Bereichen 1 und 4. Entsprechend werden zur Beschreibung dieser Situation die Regeln R1 und R5 benötigt. Ebenso ist zu  $t_1$  Konsistenzbedingung B in zwei verschiedenen Bereichen aktiv. Dies wird durch die Regeln R3 und R6 erfaßt. Die Überprüfung der Konsistenzbedingung C wird in Bereich 4 durch zwei verschiedene Ereignisse ausgelöst, was in Regel R2 vermerkt ist. Die Konsistenzbedingungen D und E sind jeweils nur in einem einzigen Bereich aktiv (Regeln R4 und R7).

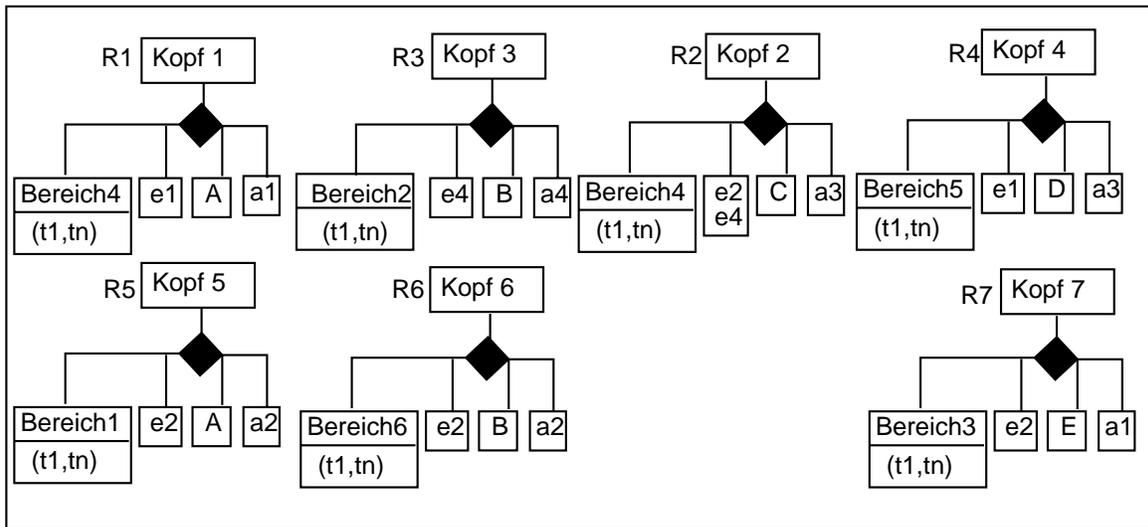
Unter der Annahme, daß der Entwurf mit dem Zeitpunkt  $t_1$  beginnt, spiegelt Abbildung 23a den Datenbasiszustand während der ersten Entwurfsphase wieder. Deshalb besitzen in Abbildung 23a alle Regeln dieselbe Zeit  $t_1$  als Anfangswert des Historie-Intervalles. Als Endzeitpunkt wird hier  $t_n$  angenommen.  $t_n$  repräsentiert einen Zeitpunkt weit in der Zukunft und sagt aus, daß die betroffene Regel momentan aktuell ist.

Zum Zeitpunkt  $t_2$  werden nun die Änderungen gemäß Abbildung 12b vorgenommen. Alle betroffenen Regeln werden entsprechend der neuen Situation angepaßt. Entsprechend unserem Vorgehen wird für alle Bereichsregeln, in denen ein Bestandteil geändert wird, im Historie-Intervall  $t_n$  durch  $t_2$  ersetzt, und für all diese Regeln wird eine Kopie angelegt, in denen die jeweiligen Änderungen vermerkt werden. Ganz allgemein können folgende Typen von Änderungen vorkommen:

- Eine Konsistenzbedingung gilt in einem neuen Bereich.
- Der Bereich einer Konsistenzbedingung ändert sich.
- Die auslösenden Ereignisse werden geändert.
- Die zugeordneten Aktionen werden geändert.
- Eine Konsistenzbedingung wird in einem Bereich ungültig.

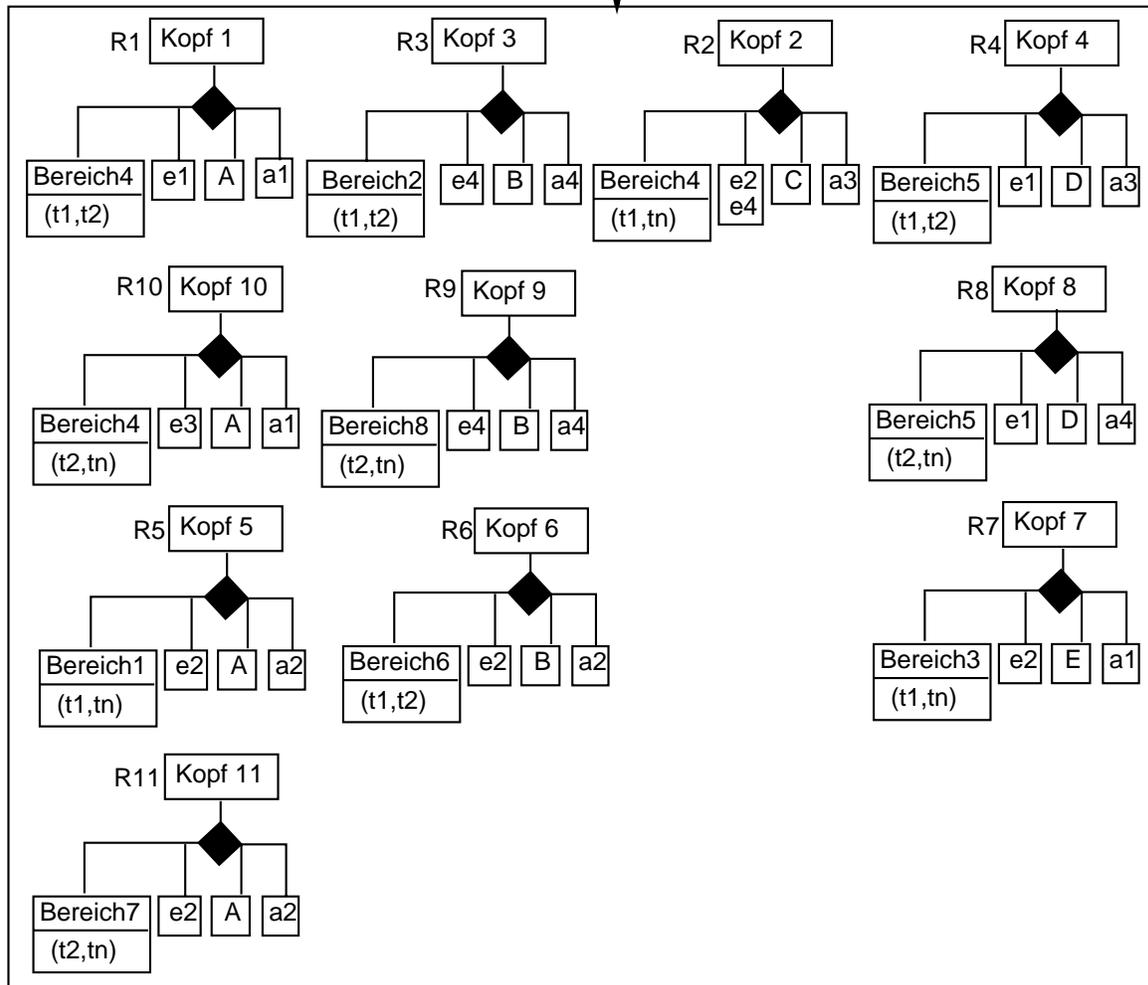
Für die Änderungen zum Zeitpunkt  $t_2$  bedeutet dies konkret (die neue Regelmengde wird in Abbildung 23b dargestellt):

- In Bereich4 wird die Konsistenzbedingung A durch ein anderes Ereignis  $e_3$  überprüft. Deshalb wird das Historie-Intervall in Regel R1 abgeschlossen, indem der Endzeitpunkt dieser Regel auf  $t_2$  gesetzt wird. Zur Erfassung des neuen Sachverhaltes wird die Regel R10 generiert, deren Anfangs-Historie-Punkt auf  $t_2$  und deren Endzeitpunkt auf  $t_n$  gesetzt wird.



a: Regeln zum Zeitpunkt t1 generiert

Änderungen zum Zeitpunkt t2



b: Regeln zum Zeitpunkt t2

Abbildung 23: Dynamische Anpassung der Regeln

- Die Konsistenzbedingung A wird zusätzlich in Bereich7 mit dem auslösenden Ereignis e2 und der Aktion a2 aktiviert. Es entsteht hierzu die neue Regel R11, deren Historie-Intervall zum Zeitpunkt t2 beginnt und als Endzeitstempel wiederum tn enthält.
- Die Konsistenzbedingung B in Bereich6 soll nun nicht mehr mit überwacht werden. Deshalb wird das Historie-Intervall für Regel R6 abgeschlossen, indem der Endzeitpunkt auf t2 gesetzt wird.
- Dafür wird der Bereich2, in dem die Konsistenzbedingung B ebenfalls gültig war, ausgeweitet. Deshalb wird nun ebenfalls die Regel R3 entsprechend der Regel R6 abgeschlossen und eine neue Regel R9 mit der Anfangszeit t2 und der Endzeit tn angelegt, in der dann auf den veränderten Bereich8 verwiesen wird.
- In Bereich5 wird bei Verletzung von Konsistenzbedingung D statt der Aktion a3 die Aktion a4 ausgelöst. Dementsprechend wird die Regel R4 analog zur Regel R1 behandelt, und es wird die Regel R8 generiert.
- R2, R5 und R7 bleiben durch die Änderungen zum Zeitpunkt t2 unverändert. Sie finden sich mit denselben Daten in der Regelmenge zum Zeitpunkt t3 wieder. Ihre Lebenszeit liegt weiterhin zwischen t1 und tn.

Im architektonischen Entwurf existieren keine feste Zeiten, zu denen Änderungen vorgenommen werden. Dementsprechend bekommt jede Regel ihre eigenen Zeitstempel, je nach dem wann der Architekt diese anlegt oder verändert.

Im Laufe des Entwurfes wächst so die Regelmenge stetig an. Dies gestattet ein beliebiges Rücksetzen in Planungszustände der Vergangenheit. Dieses Rücksetzen kann, aufgrund der Bereichsorientiertheit, in beliebigen Ausschnitten des Planes geschehen.

## 5.5 Auswerteverfahren

In diesem Abschnitt werden unsere ersten Überlegungen zu einem Auswerteverfahren skizziert. Abbildung 12 dient uns wiederum als Beispiel. Ausgangspunkt ist die Arbeitsumgebung, in der sich der Planer aktuell bewegt. Der Grundgedanke ist nun, alle Constraintcontainer, die sich mit dieser Arbeitsumgebung überschneiden, zu überwachen. Die A-Regeln, die in der Datenbank für diese Constraintcontainer stehen, werden im folgenden ÜA-Regeln (zu überprüfende A-Regeln) genannt.

Prinzipiell müssen dann die folgenden Schritte zur Überprüfung durchgeführt werden:

1. Selektiere alle ÜA-Regeln.
2. Selektiere aus der Menge aller ÜA-Regeln diejenigen aus, in dessen Bereich ein Ereignis eine Wirkung gezeigt hat.
3. Teste diese Regeln auf ihre Ereignisse.
4. Ist ein Ereignis erfüllt, so überprüfe die dazugehörige Konsistenzbedingung.
5. Ist die Konsistenzbedingung verletzt, führe die Aktionen aus.

Zur Selektion der ÜA-Regeln muß bestimmt werden, welche Bereiche sich mit der aktuellen Arbeitsumgebung schneiden. Ein Schnitt liegt vor, wenn gleichzeitig folgende Kriterien erfüllt sind:

- Die geometrischen Bereiche überlappen sich.

- Die aktuelle Systemzeit ist in den beiden Zeitintervallen und in Historie enthalten.
- Die Werte der diskreten Dimensionen entsprechen den Umgebungsvariablen der aktuellen Arbeitsumgebung.

Die verschiedenen Arten, wie ein Benutzer die Constraints überprüft haben will, werden alle durch Ereignisse erfaßt. Beispielsweise bekommt ein eine Bedingung, die prinzipiell immer überprüft wird, ein entsprechendes Ereignis “immer” zugeordnet.

Bei den Aktionen mag es häufig genügen, lediglich Meldungen an die Benutzeroberfläche abzusetzen. Denkt man bei den Aktionen auch an Kompensationsmaßnahmen, so können Rückwirkungen auf den Datenbasiszustand auftreten. In diesem Fall muß nach Ausführung einer Aktion eine erneuerte Überprüfung der ÜA-Regeln durchgeführt werden.

Zur Implementierung ist noch anzumerken, daß die Auswahl der Regeln, deren Bereich sich mit der aktuellen Arbeitsumgebung überlappt, sehr zeitkritisch ist. Diese Auswahl sollte daher durch mehrdimensionale Zugriffspfade ([2, 3]) unterstützt werden. Als Selektionskriterium dienen dann die Bereichsbeschreibungen der aktuellen Arbeitsumgebung. Verwendet man anstelle der aktuellen Arbeitsumgebung einen beliebigen vom Benutzer spezifizierten Bereich, wird es dem Benutzer ermöglicht, in beliebigen Bereichen die Konsistenz auch lokal überprüfen zu lassen.

## 6 Zusammenfassung und Ausblick

Diesem Papier liegt die These zugrunde, daß sich die Entwurfsentscheidungen mithin die Entwurfsspielräume in den früheren kreativen und damit sehr dynamischen Entwurfsphasen durch Constraints systematisch erfassen und deshalb besonders günstig mithilfe einer um Konsistenzprüfungsmechanismen erweiterten objektorientierten Datenbank unterstützen lassen. Um dies zu belegen, wurde zunächst der Entwurfsprozeß allgemein beschrieben. Anschließend wurde der A4-Raum als eine Formalisierungsmethode eingeführt, mit der sich alle Entwurfsentscheidungen einheitlich durch einen gewissen Satz von Attributen beschreiben lassen. Hierauf folgt dann die speziellen Anforderungen an einen Constraintüberwachungsmechanismus. Dabei wurden Constraints um zusätzliche Eigenschaften ergänzt:

- Positionen, die ihren Gültigkeitsbereich beschreiben,
- Ereignisse, die Aussagen darüber machen sollen, wann eine Konsistenzbedingung überprüft werden soll,
- und Aktionen, die im Falle einer Verletzung ausgeführt werden sollen.

Anschließend wurde die technische Realisierung der aufgestellten Anforderungen skizziert. Als zentrales Konstrukt wurden dazu Bereichsregeln eingeführt.

Die vorgeschlagene Konzeption bietet eine Reihe von Vorteilen:

- Architekten arbeiten bereichsorientiert. Diese Beobachtung führt zur Einführung der Positionen als zentralen Bezugspunkt für Constraints. Somit werden *Constraints bestimmten Bereichen zugeordnet* und nicht wie sonst üblich realen Objekten des Entwurfes oder bestimmten Objekttypen. Ein und dieselbe Konsistenzbedingung kann hier in vielen unterschiedlichen Bereichen gelten. Diese Bereiche (d.h. Positionen) sind keineswegs statischer Natur, die ihnen zugeordneten Attribute können jederzeit modifiziert werden.
- Die vorgeschlagene objektorientierte Implementierung erlaubt es, die eigenständig zu behandelnden Positionen, Constraints, Ereignisse und Aktionen jeweils als eigenständige Objekte zu führen. Dieses Vorgehen ermöglicht es, Bibliotheken von Ereignissen, Bedingungen

und möglichen Aktionen zu führen. Architekten erhalten hierdurch die Möglichkeit, wie in einem Puzzle, sich Ketten von Ereignissen, Bedingungen und Aktionen aus diesen *Bibliotheken* zusammenzubauen.

- Diese modulare Sichtweise auf Positionen, Constraints, auslösende Ereignisse und Aktionen ist zudem eine Voraussetzung für das Tolerieren *inkonsistenter Zustände oder die Einführung verschiedener Modi der Behandlung von inkonsistenten Bedingungen*. Diese können durch die Bereichsregeln lokal für jede Position, in der sie gelten, verwaltet werden. Die Behandlung eines Constraints im Falle einer Verletzung kann über die Aktionen, die dann auszuführen sind, gesteuert werden. Hier kann der Architekt seine ganz spezifischen Reaktionen realisieren. Diese Reaktionen können dynamisch während des Entwurfes verändert werden.
- Da Constraints Entwurfsspielräume umschreiben, ist ein *einfaches Monitoring* der Constraints unerlässlich. Unsere Konzeption erlaubt es, in
  - jedem beliebigen Ausschnitt des Entwurfes effizient zu kontrollieren, welche Constraints aktiv sind und in welchem Zustand sie sich befinden,
  - und umgekehrt für jeden Constraint ohne erneute Prüfung festzustellen, in welchen Ausschnitten des Entwurfes er gelten sollte und in welchem Zustand er sich jeweils befindet.
- Durch die Vorabdefinition der Bereichsregeln kann viel Wissen der Fachplaner beschrieben werden, das sich dann dynamisch in den Entwurf einbringen läßt.
- Rücksetzbarkeit wird durch das Konzept der “Historie” effizient unterstützt. Beim Rücksetzen müssen lediglich die Bereichsregeln gesucht werden, deren Historie den betreffenden Zeitpunkt beinhaltet. Das Vorgehen, jeder Bereichsregel ihr eigenes Attribut Historie zuzuordnen, erlaubt eine sehr *feingranulare Rücksetzbarkeit*, sowohl was den Bereich betrifft als auch die Zeitpunkte, zu denen zurückgesetzt werden soll.
- Mit der Zuordnung von Constraints zu Positionen läßt sich eine *effiziente Kontrolle von Constraints* verbinden. Die Überprüfung der Constraints kann so lokal für Objekte in den entsprechenden Bereichen durchgeführt werden, Objekte in anderen Bereichen müssen erst gar nicht überprüft werden. Auch kann man die Überprüfung auf Zeitpunkte beschränken, zu denen in den Bereichen eine Änderung stattgefunden hat.

Unsere nächsten Arbeiten werden sich nun vorrangig damit beschäftigen müssen, durch eine geeignete Realisierung des Containerservers dafür zu sorgen, daß die genannten Fähigkeiten und Eigenschaften effizient zur Verfügung gestellt werden. Auf der anderen Seite muß auch noch dafür gesorgt werden, daß der Constraintmechanismus für den Planer und Entwerfer in einer für ihn sachgerechten Weise auf der Benutzungsoberfläche verfügbar wird.

## Literatur

- [1] K. Abramowicz, B. Boss, V. Hovestadt, J.A. Mülle, R. Sturm, and P.C. Lockemann. Konsistenzüberwachung in Datenbanksystemen - Eine Anforderungsanalyse anhand der Entwurfsbereiche Architektur und Schiffbau. In *Datenbanken in Büro, Technik und Wissenschaft (BTW)*. Springer Verlag, Reihe Informatik Aktuell, 1995.
- [2] D.v. Buschmann. Mehrdimensionale zugriffspfade in einer architekturdatenbank. Universität Karlsruhe, 1993.
- [3] Gunnar Dittloff. Implementierung eines mehrdimensionalen Zugriffspfades (R-BAum) in einer Architekturdatenbank. Universität Karlsruhe, Februar 1995.
- [4] F. Haller, K. Friedrichs, V. Hovestadt, P. C. Lockemann, J. A. Mülle, and R. Sturm. The Design Navigator. In *Proc. 5th Int. Conf. Computing in Civil and Building Engineering, Anaheim, California*, pages 335–340. Computing of V-ICCCBE, 1993.
- [5] D. Henckels. Analyse eines konkreten Architektorentwurfes und Validierung der Architektur-Datenbankschnittstelle. Universität Karlsruhe, 1994.
- [6] L. Hovestadt. A4 - Digitales Bauen - Ein Modell für die weitgehende Computerunterstützung von Entwurf, Konstruktion und Betrieb von gebäuden. Universität Karlsruhe, 1994.
- [7] L. Hovestadt, V. Hovestadt, J. A. Mülle, and R. Sturm. ArchE – Entwicklung einer Datenbankunterstützten Architektur-Entwurfsumgebung. Technical Report Nr.23/94, Universität Karlsruhe, November 1994.
- [8] P. C. Lockemann, J. A. Mülle, R. Sturm, and V. Hovestadt. Modeling and integrating design data from experts in a CAAD-environment. In *In Proc. of. the European Conference on Product and Process Modelling in the Building Industry*. to appear, 1994.