

# KI-Methoden beim Entwurf komplexer Gebäude

A. Voß, C.H. Coulon, F. Gebhardt, W. Gräther, E. Groß, J.W. Schaaf, B. Schmidt-Belz

GMD, FIT

D-53754 Sankt Augustin

angi.voss@gmd.de

<http://nathan.gmd.de/projects/fabel/fabel.html>

## 1. Motivation

In diesem Artikel wird gezeigt, welche Beiträge Methoden der Künstlichen Intelligenz, insbesondere der Wissensverarbeitung, beim Entwurf komplexer Gebäude leisten können. Dies wird nicht theoretisch geschehen, sondern exemplarisch anhand des FABEL-Projekts. In FABEL untersuchten KI-Wissenschaftler, Software-Ingenieure, Psychologen und Architekten, wie wissensbasierte und fallbasierte KI-Methoden beim Bauentwurf unterstützend eingesetzt werden können<sup>1</sup> [Voß et al 96]. Der Schwerpunkt dieses Beitrags wird auf den Arbeiten in der GMD liegen, und zwar nicht auf den Details einzelner Methoden, sondern auf zentralen Entwurfsentscheidungen und Fragen der Integration.

Frühere Experimente der Architekten in FABEL hatten gezeigt, daß eine Wissensrepräsentation in Form von Objekten und Regeln bereits bei kleineren Aufgaben der Installationsplanung ihre Grenzen erreicht. Deshalb sollten in FABEL wissensintensive Methoden nur für Spezialaufgaben entwickelt werden, wie in Abschnitt 2 gezeigt wird. Daneben sollten wissensärmere, fallbasierte Methoden angeboten werden, die frühere Entwürfe direkt, ohne den Umweg über allzuviel Wissen, zur Lösung für ähnliche neue Probleme wiederverwenden. Abschnitt 3 berichtet von unseren Erfahrungen.

Die Verwendung mehrerer KI-Methoden wirft verschiedene Integrationsprobleme auf der Ebene der Wissensrepräsentation, der Datenrepräsentation, der Benutzungsoberfläche und der Gesamtsteuerung auf. Sie sind Gegenstand von Abschnitt 4. Im letzten Abschnitt 5 werden die Ergebnisse bewertet und Perspektiven aufgezeigt.

## 2. Wissensintensive KI-Methoden

Als Beispiel für eine wissensintensive KI-Methode sei AAAO vorgestellt. AAAO ist ein Akronym für Anpassung durch aktive, autonome Objekte. Die Methode plaziert Stützen in Plänen, die schon Räume oder Nutzungsbereiche enthalten. Dabei werden sowohl statische als auch architekturenspezifische und ästhetische Regeln beachtet, die für den Stahltragwerk-

---

<sup>1</sup> Diese Arbeit ist im Rahmen des Verbundvorhabens FABEL entstanden, das vom Bundesminister für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Kennzeichen "01IW 104" gefördert wurde. Die Projektpartner waren die GMD - Forschungszentrum Informationstechnik GmbH, Sankt Augustin, BSR Consulting, München, die Technische Universität Dresden, HTWK Leipzig, die Universität Freiburg und die Universität Karlsruhe.

Baukasten MIDI gelten. MIDI wurde von Fritz Haller speziell für Gebäude mit umfangreicher technischer Infrastruktur entwickelt [Haller 74]. AAAO beginnt mit einer regelmäßigen Verteilung der Stützen. Jede Stütze wird durch ein aktives autonomes Objekt repräsentiert. Die Objekte können ihre direkten Nachbarn wahrnehmen, haben Constraints, um ihre Lage zu beurteilen, und Aktionen, um sich zu bewegen, andere Objekte zu erzeugen oder sie zu löschen. Sie arbeiten im Prinzip nebenläufig, ohne eine globale Steuerung.

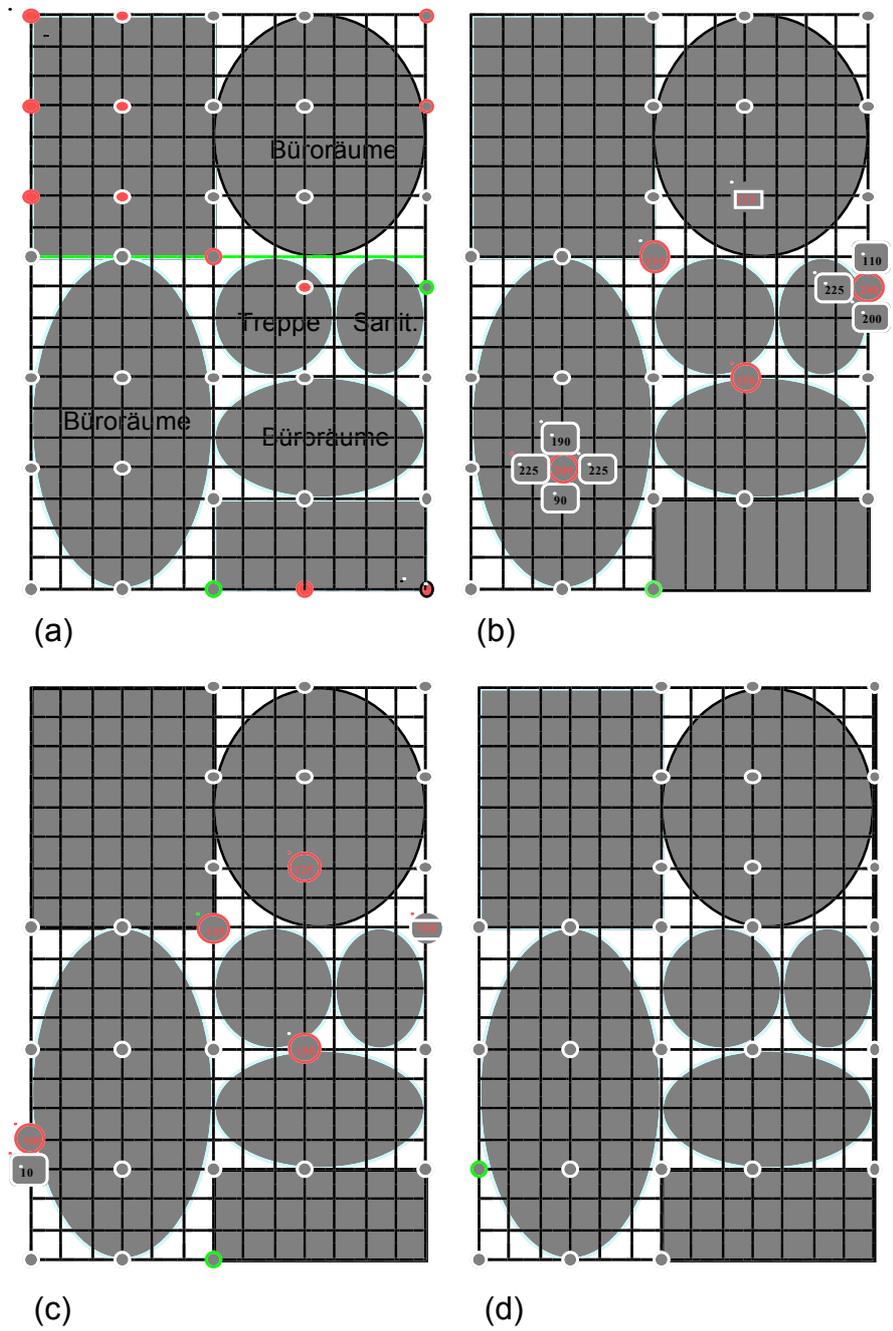


Abbildung 1: AAAO bei der Arbeit.

Abbildung 1 illustriert eine Bearbeitungssequenz. (a) zeigt den Grundriß (die Räume werden durch diejenigen Rechtecke definiert, die die grauen Ellipsen umschließen<sup>2</sup>) und eine initiale Verteilung der Stützen. (b) zeigt die erste Evaluation der Stellungen; die überflüssigen Stützen außerhalb des Gebäudes (dunkle Bereiche) und im Treppenhaus sind bereits entfernt. Die Werte gewichteten Constraintverletzungen (in Kreisen für die aktuelle Position und in Kästchen für Nachbarpositionen). (c) zeigt die Lage nach der ersten Aktion der Stützen. Eine Stütze kann sich zu einer besseren Position bewegen, während vier benachbarte Stützen sich nur durch Erzeugung einer neuen Stütze verbessern können. (d) zeigt die Endsituation, die als Lösung akzeptabel ist.

Eine weitere in der GMD entwickelte wissensintensive KI-Methode namens ANOPLA ist anwendbar auf Leitungsskizzen. Sie plant die genaue Anordnung der Leitungen mithilfe von Constraints, die aus Schablonen abgeleitet werden. Die Schablonen entstammen dem Installationsmodell *armilla* [Haller 85]. [FABEL-Report 35] enthält die Beschreibung aller wissensintensiven KI-Methoden aus FABEL.

Generell beinhaltet die Modellierung von Entwurfsaufgaben Entwurfsobjekte, Restriktionen zwischen ihnen, Anforderungen und Optimierungskriterien. Zur Bearbeitung gibt es verschiedene KI-Methoden, wie etwa Konfigurierer, Constraint-Problemlöser oder eben autonome Agenten. Will man, wie in FABEL, mehrere wissensintensive Methoden anbieten, so kann sich ihr Wissen natürlich überlappen. Es stellt sich dann die Frage, ob man das Wissen nicht für alle gemeinsam modellieren sollte. Wir haben uns aus drei Gründen dagegen entschieden: Erstens müssen die partiellen Wissensmodelle nicht unbedingt konsistent sein, vor allem die Heuristiken können sich durchaus unterscheiden. Zweitens verlangen unterschiedliche KI-Methoden unterschiedliche Repräsentationen. Zum Beispiel können Constraints als Regeln, als eigene Objekte oder lokal in Agenten dargestellt werden. Drittens wird zu jedem Zeitpunkt immer nur eine Teilmenge von KI-Methoden aktiv sein, also auch nur eine Teilmenge des Wissens im Arbeitsspeicher benötigt. In FABEL verwendet deshalb jede KI-Methode ihre spezielle Wissensrepräsentation, obwohl das zu Redundanzen führt. Die Methode ist auch verantwortlich für die Extraktion und Wartung dieses Wissens.

### 3. Wissensarme, fallbasierte KI-Methoden

Architekten lernen überwiegend an Beispielen. An der Hochschule werden sie mit Aufgaben wachsenden Schwierigkeitsgrads konfrontiert. Später erinnern sie sich an Bauprojekte, an denen sie früher beteiligt waren, oder lassen sich durch Beispiele aus Fachzeitschriften inspirieren. Diese Informationssuche kann mit fallbasierten Methoden erleichtert werden. Sie erschließen ein externes Gedächtnis in Form von elektronischen Archiven. Für einen vorgelegten Entwurfsausschnitt suchen sie ähnliche Fälle. Passende Fälle können vom Planer aufgegriffen und interaktiv oder automatisch eingepaßt werden.

#### 3.1 Retrieval

Vor der Entwicklung eines Retrieval-Verfahrens ist zu klären, was zwei Entwürfe denn ähnlich macht. Darauf gibt es bei Gebäudeentwürfen keine eindeutige Antwort. Ein Entwurf kann zum Beispiel auf seine Stückliste reduziert werden, rein optisch als Bild (d.h. als

---

<sup>2</sup> Diese ungewöhnliche Darstellung hat zwei Vorteile. Eine Ellipse sieht skizzenhafter aus als das umschließende Rechteck, und während bei den Rechtecken Kanten oder Ecken aufeinanderfallen, kann man viele überlappende Ellipsen immer noch voneinander unterscheiden.

Pixelmatrix) oder als Struktur betrachtet werden. Ein Entwurf kann auch als Ansammlung von typischen Konstellationen, sogenannten Gestalten, aufgefaßt werden, die Indizien für bestimmte Entwurfsprinzipien sind. Beispiele für Gestalten sind ringartige, kammartige, fischgrätenartige Entwürfe [Voß 94].

Für die verschiedenen Interpretationen eines Entwurfs und seiner Ähnlichkeit zu anderen Entwürfen eignen sich verschiedene KI-Methoden. Stücklisten und Gestalten lassen sich durch Schlüsselwörter darstellen, die man sehr effizient in Assoziativspeichern vergleichen kann. Alternativ eignen sich Attribut-Wert-Listen für Nearest-Neighbour-Verfahren. Pixelmatrizen verschiedener Granularität kann man mit einem Entscheidungsbaum bearbeiten, und für den exakten Vergleich von Strukturen braucht man NP-vollständige Graph-Matching-Verfahren. Sie sind in [Fabelreport-13] und [Börner et al 96] beschrieben.

Die Fallretrieval-Verfahren aus FABEL ergänzen sich gut mit den wissensintensiven Methoden, da erstere auf alle Arten von Entwurfsobjekten anwendbar und. Das wirft allerdings ein neues Problem auf: Liefern die Retrievalverfahren für dieselbe Anfrage alle das gleiche Ergebnis, und wenn nicht, welches ist richtiger - oder wenigstens besser? Tatsächlich sind die Ergebnisse durchaus unterschiedlich, aber jedes könnte unter gewissen Umständen nützlich sein. Entwerfen ist so ein komplexer Vorgang, daß in den frühen Phasen andere Fälle als später heranzuziehen sind. Am Anfang sucht man nach Inspirationen, später nach Alternativen, dann möchte man wissen, wie man weiter vorgehen kann und den Plan ergänzen kann.

Jedes der erwähnten Ähnlichkeitskonzepte fokussiert also auf einen besonderen Aspekt eines Entwurfs. Um mal das eine, mal das andere, mal eine Kombination von Aspekten zu verwenden, wurde die Retrieval-Shell ASPECT entwickelt. Mit ASPECT können beliebige, durch eine Repräsentationsfunktion und eine Distanzfunktion definierte Ähnlichkeitskonzepte anfrageabhängig gewichtet und kombiniert werden. ASPECT eignet sich insbesondere für zeitaufwendige Vergleiche. Das für ASPECT entwickelte Verfahren versucht nämlich, die Anzahl der Vergleiche zu minimieren, indem bei jedem Vergleich mit einem Fall Schlüsse über die noch möglichen Distanzen zu anderen Fällen gezogen werden. Dazu ist ein hoher Speicheraufwand erforderlich, da die relativen Distanzen der Fälle untereinander in allen Aspekten vorberechnet werden. ASPECT verwendet einen Anytime-Algorithmus: je länger er rechnet, umso präziser ist das Ergebnis. Oder umgekehrt: Anfragen nach den  $n$  ähnlichsten Fällen werden schneller berechnet als die nach den  $n+k$  ähnlichsten, ebenso muß länger warten, wer die Fälle sortiert haben möchte oder die exakte Distanz zur Anfrage wissen möchte [Schaaf 96].

In ASPECT wurden vor allem die aufwendigeren Ähnlichkeitskonzepte, wie Fälle als Pixelmatrizen und Strukturen, integriert. Um NP-vollständiges Graphmatching zu vermeiden, werden allerdings statt kompletter Graphen lediglich ihre Kantenmengen miteinander verglichen. Neben ASPECT wird der Stücklistenlistenvergleich im Assoziativspeicher ASM (M steht für memory) als besonders schnelles Verfahren separat angeboten. Da jedes Ähnlichkeitskonzept seine eigene Fallrepräsentation erfordert und jedes komplette Verfahren seine Fallbasis auf eigene Art organisiert, ist nunmehr zu klären, wie und wann diese Repräsentationen erzeugt werden sollen. Der Planer, der die Fälle erzeugt und die Fallbasis inspizieren möchte, wird natürlich nur mit einer Repräsentation konfrontiert werden. Sie sollte sich mit der im CAD-System oder der dahinterliegenden Datenbank decken. Aus dieser Ur-Repräsentation könnte sich nun jedes Retrieval-Verfahren seine eigene Fallrepräsentation bzw. Fallbasis erzeugen. Da es sich bei allen Verfahren im Prinzip um denselben Mechanismus handelt, wird er als Service vom FABEL-System angeboten.

### 3.2 Wiederverwendung von Fällen durch Anpassung

Retrieval ist nur die eine Seite der Medaille. Wenn der gefundene Fall einen hinreichend guten Entwurf enthält, möchte man Teile daraus gleich kopieren und eventuell unter geeigneten Modifikationen in den aktuellen Entwurf einsetzen. Zur Unterstützung wäre eine Einfügefunktion wünschenswert oder, methodisch gesprochen, eine Fallanpassungsfunktion.

Die einfachste Möglichkeit besteht in einer losen Kopplung von Retrieval- und wissensintensiven Methoden. Zum Beispiel kann man mit einer Retrievalmethode wie ASM Entwürfe mit Stützen suchen lassen, sie aus dem Fall in den aktuellen Plan übertragen und mit AAAO anpassen lassen. Dazu muß man AAAO statt mit einer Standard-Gleichverteilung, wie in Abbildung 1, mit der Verteilung aus dem Fall starten. Ähnlich kann man exakte Rohrverlegungspläne suchen lassen, sie in den aktuellen Plan kopieren und mit ANOPLA korrigieren lassen. Wie Abschnitt 4.2 zeigen wird, ist diese "lose Kopplung" in FABEL kein Problem, da man die verschiedenen KI-Methoden nacheinander anwenden kann.

Die lose Kopplung ist darauf beschränkt, daß man wissensbasierte Werkzeuge zum Anpassen hat. Um diese Beschränkung zu lockern, wurden in FABEL wissensarme Anpassungsverfahren entwickelt. Sie basieren auf der Idee, daß die topologische Struktur eines Entwurfs viele implizite Beschränkungen erfüllt. Indem man also die Struktur eines ähnlichen Entwurfs möglichst übernimmt, hat man viele Beschränkungen schon implizit erfüllt. Das Anpassungsverfahren TOPO ist auf Allgemeinheit ausgelegt worden. Es kann Entwurfsausschnitte mit beliebig vielen Objekten (unter zehn bis zehntausende) bearbeiten. Es erfordert als Wissen lediglich Erkennungs- und Konstruktionsfunktionen für Relationen. Erkennungsfunktionen für die Relationen zwischen Objekten dienen dazu, einen Entwurf in einen Graph zu transformieren. Für die Graphen des zu bearbeitenden Entwurfs, kurz Anfrage, und des anzupassenden Falls wird ein größter gemeinsamer Teilgraph mittels einer größten Clique berechnet. Jeder Weg im Graphen des Falls, der von dem gemeinsamen Teilgraphen ausgeht, kann in die Anfrage übertragen werden. Falls auf dem Weg bisher fehlende Entwurfsobjekte liegen, werden diese mithilfe der Konstruktionsfunktionen in der Anfrage erzeugt. Konstruktionsfunktionen gestatten es, die Geometrie des zu erzeugenden Objekts so anpassen, daß es die Relation auch in der Anfrage gilt [Coulon 95].

Abbildung 2 zeigt ein Beispiel. Anfrage und ähnlicher Fall enthalten einen Abluftschacht und ein Arrangement von Auslaßöffnungen. Im Fall sind sie bereits verbunden. Da es keine eindeutige Zuordnung gibt, wird eine zufällig ausgewählt (Pfeile in Anfrage und ähnlichem Fall). Die Wege aus dem Fall werden schrittweise übertragen, entsprechend der Numerierung. Es ist nicht immer wünschenswert, alles zu übertragen, wie die Öffnungen A und I mit ihren Verbindungen. Hier muß der Planer selektieren, das Ergebnis korrigieren, oder TOPO muß mit bereichsspezifischen Heuristiken ausgestattet werden.

## Abbildung 2 : Strukturübertragung mit TOPO.

In FABEL wurden für TOPO Erkennungsfunktionen für  $13^3$  dreidimensionale räumliche Relationen definiert. Sie bestehen aus drei zweidimensionalen Projektionen, von denen es je 13 gibt. Dadurch, daß die 13 Relationen generisch sind (überlappt, berührt, enthält,...), reicht diese Modellierung für alle in FABEL auftretenden Arten von Entwürfen. Mit anderen Erkennungs- und Konstruktionsfunktionen ist TOPO aber auch auf andere Aufgaben, wie etwa in der Stadtplanung, anwendbar. TOPO gewinnt seine Allgemeinheit durch zusätzliche Benutzerinteraktionen. Wie im Beispiel erwähnt, ließen sie sich durch Hinzufügen von mehr Wissen reduzieren. Dann aber würde das Einsatzgebiet von TOPO entsprechend eingeschränkt.

## 4. Integration

Die Integration mehrerer Entwurfsmethoden in ein interaktives System wirft Probleme auf der Ebene der Wissens- und Fallrepräsentation, der Datenrepräsentation, der Benutzungsoberfläche und der Ablaufsteuerung auf.

### 4.1 Wissens- und Fallrepräsentation

Die Frage nach einer gemeinsamen Wissensrepräsentation wurde in Abschnitt 2 für die wissensintensiven Methoden aus mehreren Gründen negativ beantwortet. Für die wissensarmen Methoden wurde in Abschnitt 3.2 erläutert, daß es an der Schnittstelle zum Planer einheitliche Fallbasen und Fallrepräsentationen geben muß, daß ihre Transformation in methodenspezifische Fallbasen aber als Systemservice angeboten werden kann.

### 4.2 Objektkatalog und Datenrepräsentation

Die Frage nach einer für alle KI-Methoden gemeinsamen Datenrepräsentation ist in dem Augenblick einfach beantwortbar, da die Entscheidung für ein Unterstützungssystem fällt. In einem solchen System können die KI-Methoden nur als Werkzeuge angeboten werden, derer

sich der Planer nach Belieben bedienen mag. Die für alle KI-Methoden verbindliche Datenrepräsentation ergibt sich, wie bei den Fällen, aus der des CAD-Systems beziehungsweise der dahinterliegenden Datenbank. Alle KI-Methoden müssen Ausschnitte des Gebäudemodells aus der Datenbank bearbeiten können und ihre Ergebnisse dort ablegen können. Etwaige Transformationen sind Privatangelegenheit der jeweiligen Methode. Zum Beispiel erzeugt AAAO als erstes für jede Stütze einen Agenten und berechnet aus den Objektkoordinaten ihre Nachbarn.

Aus der gemeinsamen Datenrepräsentation ergibt sich eine hohe Anforderung an die Qualität der Daten. Die Werkzeuge sind umso besser, je semantisch haltvoller die Daten sind. In FABEL haben wir deshalb nicht auf einer Zeichnungsrepräsentation (Vektorgrafiken) aufgesetzt, sondern auf einem Objektmodell. Ein Entwurf besteht demnach aus einer Menge von (Tausenden von) Objekten. Der vom Karlsruher Partner aufgestellte Katalog umfaßt 250 Objekttypen, und zwar nicht nur konkrete, bestellbare, sondern auch abstrakte Objekte. Solche Objekte werden in den frühen Entwurfsphasen benutzt, um die ungefähre Lage von später hinzuzufügenden konkreten Bauteilen zu skizzieren, wie etwa die Ellipsen in Abbildung 1.

Die Objektdarstellung im Gebäudemodell enthält pro Objekt die Koordinaten, den Typ im Katalog und zusätzliche semantische Attribute: die Abstraktion (Hüllquader, exakt, konkretes Bauteil), das Gewerk oder Subsystem (Raum, Fassade,...), die Morphologie oder den Zweck (Erschließung, Verbindung, Nutzung) und die Größenordnung oder Skala.

Die Anreicherung der Objekte durch semantische Attribute hat zwei entscheidende Vorteile: Erstens beinhalten sie Wissen, das die KI-Methoden unmittelbar benutzen können, zweitens stellen sie eine Abstraktion von den konkreten Objekten dar. Da alle in FABEL entwickelten KI-Methoden nur auf den geometrischen und semantischen Attributen operieren, sind sie unabhängig von dem verwendeten Objektkatalog. Die KI-Methoden können über beliebigen anderen Katalogen arbeiten, solange die Objekte sich auf die genannten Attribute abbilden lassen. Darüberhinaus lassen sich die KI-Methoden aus FABEL in gewissen Grenzen auf andere semantische Attribute übertragen. Wichtig ist allerdings, daß es ein gemeinsames Gebäudemodell gibt (oder auch verschiedene Varianten und Versionen), auf dem alle am Planungsprozeß beteiligten Personen und Werkzeuge aufsetzen.

### **4.3 Das Planungsmodell**

Die Einführung eines Objektkatalogs und der semantischen Attribute löst zum Teil das Problem, wann im Planungsprozeß welche Werkzeuge einsetzbar sind. Denn jedes wissensintensive Werkzeug kann bestimmte Arten von Entwurfsobjekten bearbeiten. Es ist also anwendbar auf solche Ausschnitte des Gebäudemodells, die die vorausgesetzten Objekte bereits enthalten, aber noch nicht die von dem Werkzeug gegebenenfalls zu erzeugenden Objekte. Diese Definition ist zwar formal klar, aber nicht unbedingt anschaulich. Wie kann man auf einen Blick sehen, ob sich ein Werkzeug für den betrachteten Ausschnitt eignet?

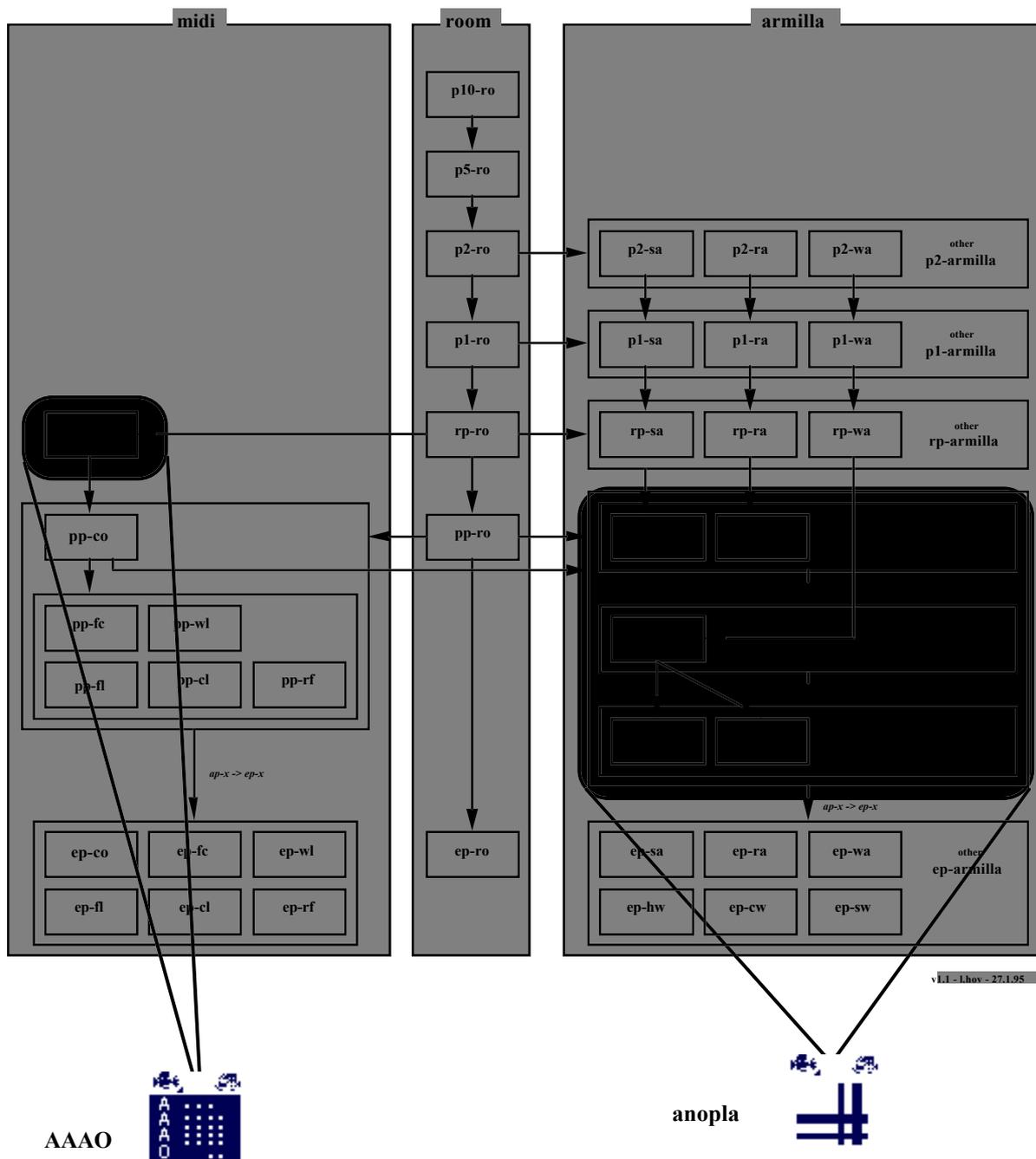


Abbildung 3: Ausschnitt aus dem Planungsmodell mit dem Einsatzbereich der Werkzeuge AAAO und ANOPLA.

In FABEL wurde ein sogenanntes Planungsmodell PM5 entwickelt, das den Bauentwurfsprozeß in Teilaufgaben zerlegt. Über solche Modelle mögen die Fachleute streiten und jeder ein anderes Modell favorisieren, entscheidend ist, daß die Teilaufgaben bestimmte Typen von Entwurfsobjekten erzeugen und daß Abhängigkeiten zwischen Teilaufgaben so definiert werden können, daß die neuen Objekte in Abhängigkeit von den Objekten geplant werden können, die in den vorausgesetzten Teilaufgaben entstanden sind. Diese Aufgabenzerlegung dient nicht etwa einer vollautomatischen Steuerung, sondern der informellen Orientierung des Planers. Man kann sie als Karte darstellen und darin die Gebiete einzeichnen, für die KI-Werkzeuge zur Verfügung stehen. Dabei handelt es sich vornehmlich um wissensintensive Methoden, da die wissensärmeren, fallbasierten Methoden tendenziell aufgabenübergreifend einsetzbar sind. Abbildung 3 zeigt einen vereinfachten Ausschnitt aus

dem Planungsmodell von FABEL und die den Werkzeugen AAAO und ANOPLA zugeordneten Wirkungsbereiche [Schmidt-Belz, Hovestadt 96]. Die Präfixe p, pp und ep stehen für Planungsstufen auf den drei Abstraktionsebenen, die Suffixe bezeichnen verschiedene Subsysteme, wie co für Konstruktion, rf für Dach, cw für Kaltwasser.

### **4.3 Die Metapher der virtuellen Baustelle**

Das Gebäudemodell ist in einer Datenbank abgelegt. Die Interaktion zwischen dem Planer und den Werkzeugen verläuft so, daß der Planer in seinem CAD-System einen Ausschnitt aus dem Gebäudemodell auswählt und bearbeitet. "Bearbeiten" heißt, daß er manuell Objekte hinzufügt, löscht oder verändert oder daß er einen Teilausschnitt von einem Werkzeug bearbeiten läßt. Während das KI-Werkzeug aktiv ist, kann der Planer weitere Teilausschnitte bearbeiten oder von wiederum Werkzeugen bearbeiten lassen. Zum Beispiel könnte der Planer zunächst einige Abluftöffnungen als Anfrage für ASM oder ASPECT selektieren, um geeignete Fälle für das Anpassungswerkzeug TOPO zu finden. Während TOPO noch operiert, kann der Planer den Vorgang an anderer Stelle wiederholen. Und während die später gestarteten Instanzen von TOPO noch aktiv sind, könnten die früher gestarteten schon die ungefähre Lage der Abluftleitungen eingetragen haben. Diese könnte der Planer mit dem Werkzeug ANOPLA bearbeiten lassen, um präzise platzierte Leitungen einzufügen. Unterdessen werden weitere Instanzen von TOPO fertig usw. Da dem Planer die Gesamtsteuerung obliegt, muß er auch darauf achten, daß es nicht zu Kollisionen zwischen den Werkzeugen und ihren Wirkungsbereichen kommt. Da aber die Werkzeuge ihre Ergebnisse als Angebot präsentieren und ablegen, die der Planer annehmen, ignorieren oder löschen kann, besteht keine Gefahr, daß der Entwurf unkontrolliert überschrieben wird.

Ein solches Vorgehen setzt voraus, daß der Planer die Übersicht über den Stand des Entwurfs und die aktiven Werkzeuge behält. Um dies zu gewährleisten, hat sich FABEL der Metapher der virtuellen Baustelle bedient. Heutige CAD-Systeme gestatten es, Sichten auf den Gesamtentwurf zu legen, durch den Entwurf zu navigieren oder ihn sogar dreidimensional zu "begehen", ähnlich wie ein echtes Gebäude. Auf einer echten Baustelle sieht man aber nicht nur das teilweise fertige Gebäude, sondern auch die Werkzeuge und die Arbeiter, und bekommt so einen Überblick über den Prozeß. Ähnlich kann der Planer auf der virtuellen Baustelle, so wie sie FABEL präsentiert, nicht nur die Entwurfsobjekte sehen, sondern auch die dazwischen platzierten aktiven oder passiven Werkzeuge. Die Werkzeuge werden wie Entwurfsobjekte platziert und wie diese mit ihren Koordinaten in der Datenbank abgelegt.

### **4.4 Der FABEL-Prototyp**

Im FABEL-Prototyp wurde die virtuelle Baustelle über ein Rechnernetz realisiert, in dem mehrere Prozesse für die jeweiligen Werkzeuge ablaufen. Idealerweise sollte die virtuelle Baustelle als Erweiterung eines kommerziellen CAAD-Systems implementiert sein. Die Architekten in FABEL arbeiteten mit Minicad, das aber leider nicht offen ist und nur die üblichen, eingeschränkten Navigationsmöglichkeiten bietet. Darum tauschen wir mit Minicad lediglich Daten für die professionelle Bearbeitung aus und benutzen zusätzliche Forschungsprototypen des Karlsruher Partners, A5Broker als Server für das Gebäudemodell und A5Draw als objektorientiertes Zeichenwerkzeug. Beide laufen unter dem Betriebssystem NeXTStep. A5Draw unterstützt die Navigation durch die virtuelle Baustelle auf komfortable Weise und ist offen für die Anbindung von Werkzeugprozessen im Hintergrund. Beides sind notwendige Voraussetzungen zur Umsetzung der Metapher der virtuellen Baustelle.

Das FABEL-System läßt sich dynamisch konfigurieren. Die Minimalversion besteht aus A5Draw und A5Broker. Beide können auf einem einzigen Rechner laufen und über das

TCP/IP-Protokoll kommunizieren. Natürlich können A5Draw und A5Broker auch auf verschiedene Rechner verteilt werden und in mehreren Instanzen vertreten sein, damit die Planer verteilt arbeiten können. Dieses Rahmensystem wurde durch die KI-Werkzeuge ergänzt. Jedes KI-Werkzeug bietet in seinem Wirkungsbereich auf der virtuellen Baustelle ein Panel mit seinen Hauptfunktionen an. Die Werkzeuge selbst sind meist in Lisp implementiert und können unter Unix auf einem oder mehreren Rechnern verteilt ablaufen. Sie kommunizieren mit den Panelen über TCP/IP. Der Prototyp läuft seit Dezember 1995 [Gebhardt et al 96].

## **5. Ausblick**

Viele Fragen, mit denen wir uns bei der Entwicklung des FABEL-Prototypen konfrontiert sahen und die in diesem Beitrag angesprochen wurden, sind in dieser Form noch nie gestellt worden, da noch niemand eine ähnliche Vielfalt von KI-Methoden für den Entwurf komplexer Gebäude in einem System integriert hat. Damit hat der Prototyp seinen Zweck erfüllt. Er sollte nunmehr als Steinbruch dienen, um ausgewählte FABEL-Methoden in hoffentlich bald verfügbare offene kommerzielle CAD-Systeme zu integrieren. Unter diesen Methoden findet man wissensintensive Spezialprogramme, inhaltsgesteuerte Retrievalverfahren und kontextsensistive Verfahren zur Wiederverwendung von CAD-Plänen. Solche Methoden dürften auch interessant werden für den Zugriff auf und die Wiederverwendung von elektronisch im Internet angebotenen Entwürfen.

Viele unserer Erkenntnisse dürften sich auf ähnliche Aufgabenstellungen verallgemeinern lassen. Damit ist der Entwurf von komplexen Einmalprodukten gemeint, wie von Gebäuden, Städten oder Schiffen. Insofern die Produkte länger existieren als das zu ihrer Konstruktion herangezogene Wissen gültig ist und insofern die Entwürfe elektronisch archiviert sind, muß bei der allfälligen Umplanung das in den Entwürfen vorhandene Know-how stärker reaktiviert werden. Dazu eignen sich fallbasierte Methoden [Voß 96]. Einen guten Überblick bietet [Gebhardt et al 97].

Zuguterletzt hat die Formalisierung der Baudomäne, die Einführung des Planungsmodells und des Objektkatalogs für das gemeinsame Gebäudemodell zu einer neuen Qualität der Datenrepräsentation geführt. Die in FABEL geplanten Gebäude sind absolut präzise. Sie dienen mittlerweile als "Musterhäuser". Durch manuelles Navigieren gemäß der Aufgabenstruktur begibt sich der Planer zu den als nächstes zu beplanenden Stellen, und zwar sowohl im aktuellen Entwurf wie auch im Musterentwurf. Dort kopiert er die entsprechenden Objekte, fügt sie in den neuen Entwurf ein und modifiziert sie gegebenenfalls manuell. Auf diese Weise lassen sich einfacher und schneller detailliertere Entwürfe erstellen, so daß das hektisches Um- oder Zu-Ende-Planen auf der echten Baustelle mit den verbundenen Kosten deutlich reduziert werden kann.

## **Literatur**

[Börner et al 96] Börner, Katy, Pippig, Eberhard, Tammer, Elisabeth-Ch. and Coulon, Carl-Helmut: Structural similarity and adaptation. In (Ian Smith and Boi Faltings eds), *Advances in Case-Based Reasoning*, EWCBR-96, pp. 58-75, Springer, Berlin, 1996.

[Coulon et al 96] Coulon, Carl-Helmut, Gräther, Wolfgang, Schmidt-Belz, Barbara, Voß, Angi, Gebhardt, Friedrich, Groß, Eckehard, Schaaf, Jörg: *Virtual building site: supporting*

building design by multiple methods in FABEL. In (John S. Gero and Fay Sudweeks eds.) Artificial Intelligence in Design'96, pp. 465-483, Kluwer Academic Publishers, Dordrecht, 1996.

[Coulon 95] Coulon, Carl-Helmut: Automatic Indexing, Retrieval and Reuse of Topologies in Complex Designs. In (Peter Jan Pahl and Heinrich Werner eds.) Proceeding International Conference on Computing in Civil and Building Engineering, pp. 749-754, Berlin, 1995. Balkema Verlag, Rotterdam.

[Fabelreport-13] Voß, Angi: Similarity concepts and retrieval methods, Fabel-Report 13, GMD, Sankt Augustin, Juni 1994.

[Fabelreport-35] Börner, Katy: Modules for design support, Fabel-Report 35, GMD, Sankt Augustin, Juni 1995.

[Gebhardt et al 97] Gebhardt, Friedrich, Voß, Angi, Gräther, Wolfgang, Schmidt-Belz, Barbara: Reasoning with complex cases. Kluwer, Boston, 249 pages, to appear.

[Haller 74] Haller, Fritz: MIDI - ein offenes system für mehrgeschossige bauten mit integrierter medieninstallation, 1974, Münsingen, USM baustysteme haller.

[Haller 85] Haller, Fritz: ARMILLA - ein installationsmodell, 1985, Institut für industrielle Bauproduktion, Universität Karlsruhe.

[Schaaf 96] Schaaf, Jörg W.: Fish and Shrink: a next step towards efficient case retrieval in large scaled case bases. In (Ian Smith and Boi Faltings eds), Advances in Case-Based Reasoning , EWCBR-96, pp. 362-376, Springer, Berlin, 1996.

[Schmidt-Belz, Hovestadt 96] Schmidt-Belz, Barbara Hovestadt, Ludger: Scenario of an integrated design support for architects. In Design Studies, 1996 (to appear).

[Voß 94] Voß, Angi : Retrieval of similar layouts in FABEL. In IKM, Abstracts, pp.163-168, Hochschule für Architektur und Bauwesen, Weimar, März 1994.

[Voß et al 96] Voß, Angi, Bartsch-Spörl, Brigitte, Hovestadt, Ludger, Jantke Klaus P., Petersohn, Uwe , Strube, Gerhard: FABEL. In Künstliche Intelligenz 10(3):70-76, 1996.

[Voß 96] Voß, Angi: Principles of case reusing systems. In (Ian Smith and Boi Faltings eds), Advances in Case-Based Reasoning , EWCBR-96, pp. 428-444, Springer, Berlin, 1996.