



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

**Research
Report**
RR-91-35

**Constraint-basierte Verarbeitung
graphischen Wissens**

Winfried Graf, Wolfgang Maaß

Oktober 1991

**Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH**

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

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

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

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Intelligent Communication Networks
- Intelligent Cooperative Systems.

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

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

Prof. Dr. Gerhard Barth
Director

Constraint-basierte Verarbeitung graphischen Wissens

Winfried Graf, Wolfgang Maaß

DFKI-RR-91-35

Diese Arbeit ist eine überarbeitete Version des in
W. Brauer, D. Hernández (Hrsg.): *Verteilte Künstliche Intelligenz und kooperatives Arbeiten*,
Proceedings 4. Internationaler GI-Kongreß Wissensbasierte Systeme, München, Oktober 1991,
IFB 291, Springer-Verlag, Berlin, 1991
erschienenen Artikels.

Diese Arbeit wurde finanziell unterstützt durch das Bundesministerium für Forschung und
Technologie (FKZ ITW-8901 8).

© Deutsches Forschungszentrum für Künstliche Intelligenz 1991

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

Constraint-basierte Verarbeitung graphischen Wissens

Winfried Graf und Wolfgang Maaß

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)

Stuhlsatzenhausweg 3

W-6600 Saarbrücken 11

e-mail: {graf,maass}@dfki.uni-sb.de

Zusammenfassung

Bei der Entwicklung neuerer intelligenter Benutzerschnittstellen, die wie im Beispiel des multimodalen Präsentationssystems WIP natürliche Sprache und Graphik kombinieren, spielt insbesondere die wissensbasierte Gestaltung des Layouts multimodaler Dokumente eine wichtige Rolle. Am Beispiel des Layout-Managers in WIP soll gezeigt werden, wie aufgrund der von einem Präsentationsplaner spezifizierten semantischen und pragmatischen Relationen, die von den media-spezifischen Generatoren erzeugten Graphik- und Textfragmente in einem Dokument automatisch arrangiert werden können. Dabei wird das Layoutproblem als Constraint-Satisfaction-Problem behandelt. Es wird hier gezeigt, wie der Constraint-Ansatz sowohl zur Repräsentation von graphischem Wissen, als auch zur Berechnung der Platzierung der Layoutobjekte auf einem Design-Grid verwendet werden kann. So werden semantische Kohärenzrelationen wie etwa *'sequence'* oder *'contrast'* durch entsprechende Design-Constraints reflektiert, die perzeptuelle Kriterien (Alignierung, Gruppierung, Symmetrie, etc.) spezifizieren. Zur Realisierung wird in WIP ein mehrschichtiger inkrementeller Constraint-Solver mit lokaler Propagierung verwendet, der es erlaubt, Constraints dynamisch zu generieren.

Inhaltsverzeichnis

1	Einleitung	3
2	Stand der Forschung	4
3	Automatisches Layout	5
4	Repräsentation von graphischem Wissen	6
5	Ein mehrschichtiger Constraint-Solver zur Verarbeitung graphischen Wissens	8
5.1	Verarbeitung lokaler Beziehungen	9
5.2	Verarbeitung globaler Beziehungen	10
6	Resümee	11

1 Einleitung

Das Design bildet den zentralen Punkt bei vielen komplexen menschlichen Tätigkeiten, wie z.B. auch der Gestaltung des Layouts von multimodalen Präsentationen. Dabei wollen wir in dieser Arbeit unter dem Design von Dokumenten die Bestimmung einer adäquaten äußeren Form verstehen, die gewünschten ästhetischen Charakteristika genügt und darüberhinaus gewisse funktionale Anforderungen erfüllt. Hier zählt nicht der spektakuläre Entwurf, sondern die systematische Konzeption des Layouts nach grundlegenden Gestaltungskriterien.

Psychologische Untersuchungen haben gezeigt, daß eine klare und logische Gestaltung der Form einer Präsentation ihre *Effizienz* und *Expressivität*, d.h. Lesbarkeit, Verständlichkeit und Glaubwürdigkeit, erheblich fördern. Auf der Graphikseite unterscheiden wir nach [33] *funktionales* vs. *künstlerisches* Layout. Wir wollen uns bei den nachfolgenden Betrachtungen ausschließlich auf das beim Entwurf von multimodalen Dokumenten (Graphik und Text) relevante funktionale Layout konzentrieren. Hier steht insbesondere die Übermittlung des Inhalts im Vordergrund, während beim künstlerischen Layout im wesentlichen die äußere Form der Darstellung dominiert. Ein gutes Layout zeichnet sich dabei vor allem durch seinen transparenten, sachlichen, funktionalen und ästhetischen Charakter aus (vgl. [8]).

Der menschliche Designer steht heute unter dem ständigen Druck der Erhöhung der Produktivität durch immer innovativere und effizientere Produkte und erhofft sich deshalb durch den Einsatz von neuen wissensbasierten Techniken eine spürbare Entlastung. Die meisten der bisherigen Systeme zur Automatisierung des graphischen Designs bezogen sich jedoch im wesentlichen auf die Unterstützung von Routine-Designaufgaben, für die bereits eine Vielzahl von Designheuristiken bekannt war und die somit eine effektive Kontrolle des Designraums ermöglichten. In letzter Zeit rückten dann zunehmend Systeme zum konzeptionellen Design stärker in den Vordergrund. Die überwiegende Mehrheit dieser Systeme versteht dabei den Designprozeß als Kombinatorikproblem in einem diskreten Suchraum (vgl. auch [29]).

Wir wollen im weiteren das als Such- oder Optimierungsproblem vorliegende Layoutproblem als Constraint-Satisfaction-Problem (*CSP*) formulieren und mittels Konsistenztechniken lösen. Dabei werden wir zeigen, daß sich hier Constraint-Formalisten sehr gut sowohl zur deklarativen Wissensrepräsentation als auch als Berechnungsmodell und effiziente Kontrollstrategie eignen. Dieser modellbasierte Ansatz ermöglicht es, sich im Gegensatz zu bisherigen regel-basierten Ansätzen, durch die Einbeziehung von strukturellem Wissen über den Layoutprozeß, auf einige wenige Heuristiken in Form von kompiliertem Wissen zu beschränken. Dadurch kann graphisches Wissen unabhängig von domänenspezifischen Designstrategien verarbeitet werden. Wir werden insbesondere zeigen, wie sich semantische Zusammenhänge in multimodalen Dokumenten in graphische *Gestaltprinzipien* (vgl. z.B. [3]) geeignet abbilden lassen.

Im Bereich der intelligenten Benutzerschnittstellen besteht derzeit, bedingt durch die wachsende Komplexität der von wissensbasierten Anwendungssystemen zu übermittelnden Information, ein zunehmender Bedarf an Werkzeugen zur flexiblen und effizienten Informationspräsentation, deren Layout in Abhängigkeit des Benutzers und der Situation

variiert. Ziel dieser Arbeit ist es, am Beispiel des für das multimodale Präsentationssystem WIP (vgl. [39, 40]) entwickelten Layout-Managers, einen neuen constraint-basierten Formalismus zur Verarbeitung von design-relevantem graphischen Wissen vorzustellen.

2 Stand der Forschung

Das Problem der Bestimmung eines ästhetisch optimalen Layouts eines multimodalen Dokumentes unter vorgegebenen Randbedingungen, wie etwa Platzrestriktionen, Ausgabemedium, Benutzer, etc. ist NP-vollständig (vgl. [5]).

Bisherige Arbeiten im Bereich der graphischen Benutzerschnittstellen bezogen sich in erster Linie auf die Entwicklung von interaktiven 3D-Graphikeditoren. Hierbei wurde dem Benutzer eine Fülle an Graphiktechniken angeboten, die es ihm erlauben sollten, das Design effizienter als mit konventionellen Methoden zu gestalten. Da der Benutzer jedoch in der Regel nicht über das Wissen eines erfahrenen Graphik-Designers verfügt, konnte er die Möglichkeiten solcher Systeme nur schwer ausschöpfen. Außerdem standen ihm zumeist nur sehr rudimentäre Mittel zur Layoutunterstützung zur Verfügung.

In späteren Ansätzen wurde dann eine regel-basierte Unterstützung des Benutzers verfolgt. Da diese Systeme jedoch nur die zu präsentierende Information betrachten, nicht aber das Benutzerspektrum, die Präsentationssituation oder Zeitaspekte, untersuchen aktuelle Ansätze die automatische Planung des Inhalts und der Form von graphischen Präsentationen (vgl. insb. die Systeme *VIEW* [14], *APT* [21] und *APEX/IBIS* [10, 32]). Ein erster Ansatz zum automatischen Layout von Bildschirmpräsentationen ist das von Feiner entwickelte System *GRIDS* [11]. Hier wird jedoch die Generierung der Form unabhängig von der Planung des Inhalts behandelt.

Neben rein graphischen Präsentationen erfährt das automatische Design von multimodalen Präsentationen in jüngster Zeit zunehmende Aufmerksamkeit innerhalb der KI-Forschung, da in vielen Situationen eine einfache und effiziente Informationspräsentation nur durch eine Kombination verschiedener Kommunikationsmodi, wie z.B. Text, Graphik, Animation und Gestik, möglich ist. In vielen Anwendungen sind häufig sehr heterogene Arten von Information, wie z.B. räumliche Lokationen, kausale und zeitliche Beziehungen, Intentionen, konditionale Aktionen oder disjunktive Alternativen, zu übermitteln (vgl. [4]). Hierbei spielt dann insbesondere die koordinierte Generierung solcher multimodaler Präsentationen eine wichtige Rolle, die eine parallele Verwendung der verschiedenen Modi unter Ausnutzung ihrer individuellen Stärken erlaubt. Konkurrierende Ansätze bilden derzeit die Systeme *COMET* [12], *FN/ANDD* [25, 24], *SAGE* [31] und *WIP* [39, 40] sowie die Arbeiten von Hovy et al. am ISI [2].

Die Automatisierung des geometrischen Layouts zählt zu den frühesten Anwendungsfeldern für constraint-basierte Sprachen und Systeme (vgl. [18]). Eine Pionierarbeit auf diesem Gebiet stellt das von Sutherland 1963 am MIT entwickelte System *Sketchpad* [36] dar. Basierend auf diesem Ansatz folgten weitere Arbeiten von Borning et al. im Rahmen des *ThingLab*-Projektes bei Xerox PARC (vgl. [6]). Constraint-basierte Ansätze zum geometrischen Layout liegen auch den Systemen *Magritte* [15], *Juno* [30] und *IDEAL* [38] zugrunde.

3 Automatisches Layout

Bei der automatischen Generierung multimodaler Präsentationen unterscheiden wir zwischen der Planung des Inhalts und der Wahl der verschiedenen Modi sowie der Bestimmung der äußeren Form des Dokumentes. Eine zentrale Komponente in der hierarchisch organisierten Kaskadenarchitektur von WIP (**W**issensbasierte **I**nformations**p**räsentation) nimmt somit neben einem Präsentationsplaner (vgl. [1]) der Layout-Manager (vgl. [16, 19]) ein.

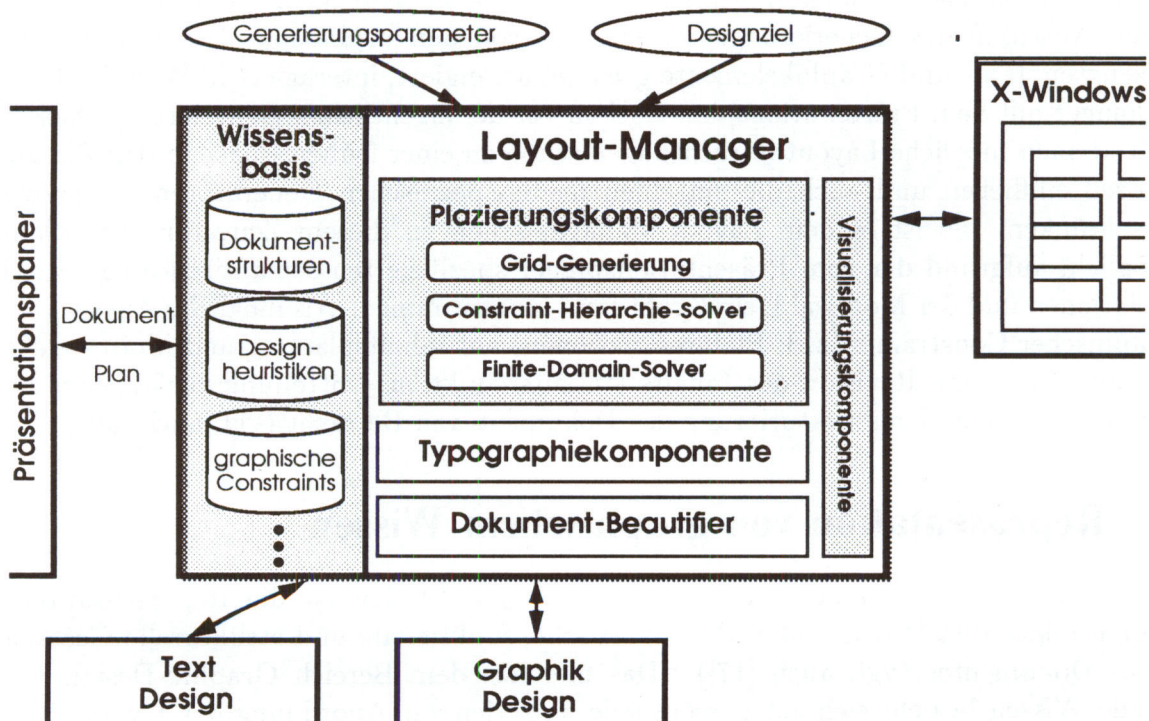


Abbildung 1: Architektur von WIP's Layout-Manager

Während der Präsentationsplaner die Selektion der zu präsentierenden Inhalte und die Wahl der Modi vornimmt, besteht die wesentliche Aufgabe des Layout-Managers darin, aufgrund der vom Präsentationsplaner spezifizierten semantischen und pragmatischen Relationen, die von den media-spezifischen Generatoren erzeugten Text- und Graphikfragmente auf einem Dokument zu arrangieren, d.h. es sind Größe und Position der einzelnen Boxen zu bestimmen.

Der Layoutprozeß wird auf zwei unterschiedlichen Detaillierungsebenen durchgeführt. Zunächst wird ein Groblayout auf der Basis von Defaultannahmen über Dokumentstrukturen sowie Designheuristiken bestimmt. Anschließend wird mittels zweier dedizierter Constraint-Solver für die instanziierten Layoutobjekte in mehreren Phasen deren endgültige Platzierung berechnet (vgl. Kap. 5). Zur Einschränkung des Dokumentraums verwenden wir hierbei das aus dem Bereich Graphik-Design bekannte Konzept des typographi-

schen Rasters (*Grid*, vgl. [28]). Dieses garantiert ein effizientes (d.h. uniformes, kohärentes und konsistentes) Design funktionellen Layouts.

Die Architektur des Layout-Manager in WIP (s. Abb. 1) umfaßt neben der im folgenden beschriebenen Platzierungskomponente (vgl. auch [19]) eine regel-basierte Komponente zur Gridgenerierung (vgl. [20]), eine automatische Typographie-Komponente sowie ein Modul zum automatischen Rendering des multimodalen Dokumentes.

Im Gegensatz zu anderen Ansätzen (vgl. [12]), in dem die Layoutkomponente erst nach Abschluß des Generierungsprozesses die von den media-spezifischen Generatoren erzeugten Text- und Graphikelemente geeignet arrangiert, interagiert in WIP der Layout-Manager mit dem Präsentationsplaner schon vor der eigentlichen Generierungsphase. Somit können mögliche Layoutbetrachtungen schon zu einer frühen Phase in den Planungsprozeß einfließen und auch die einzelnen media-spezifischen Generatoren entsprechend beschränken. So ist es dem Layout-Manager schon zu diesem Zeitpunkt der Planung möglich, aufgrund der vom Präsentationsplaner spezifizierten semantisch-pragmatischen Relationen und der Modiwahl, ein Groblayout zu bestimmen. Als mögliche Auswirkungen graphischer Constraints (z.B. Platzrestriktionen) auf Inhaltsplanung und Modusselektion kommen u.a. eine Revision des Inhalts (Inhaltsreduktion, Mitteilung zusätzlicher Information) und eine Umstrukturierung des Dokumentes in Betracht (vgl. auch [39]).

4 Repräsentation von graphischem Wissen

Ein zentrales Problem beim automatischen Layout-Design ist die Repräsentation des design-relevanten Wissens über die ästhetische, funktionale und strukturelle Gestaltung eines Dokumentes (vgl. auch [17]). Das hier aus dem Bereich Graphik-Design vorliegende Wissen bezieht sich auf perzeptuelle Kriterien zur Anordnung der Layoutobjekte, wie z.B. horizontale vs. vertikale Ordnung, Alignierung, Gruppierung, Symmetrie oder Ähnlichkeit (vgl. u.a. [3, 28, 8, 33]).

Dieses Wissen liegt jedoch größtenteils in Form von Heuristiken und Intuitionen vor und ist daher häufig unstrukturiert und unformalisiert. Darüberhinaus haben empirische psychologische Studien gezeigt, daß es eine sehr hohe Variabilität in den für eine Präsentation verwendeten Layoutmustern bei der Anordnung von Text-Bild-Kombinationen gibt (vgl. [41]). D.h. zur Modellierung des Layout-Prozesses ist neben domänenspezifischem Wissen auch *Common-Sense-Wissen* zu repräsentieren.

Die zentrale Idee des hier vorgestellten Ansatzes, besteht in einer möglichst natürlichen Kodierung der perzeptuellen Merkmale durch die Einbeziehung von strukturellem Wissen. Ein Teil des für die Bestimmung des Layouts relevanten Designwissens (z.B. zur Beschreibung von Topologie und Geometrie) läßt sich dabei mittels Constraint-Netzen (vgl. [22, 27]) deklarativ beschreiben. Constraints repräsentieren hierbei sehr heterogene Beziehungen zwischen den einzelnen Layoutobjekten.

Zur Beschreibung der verschiedenen graphischen Wissensarten unterscheiden wir numerische und symbolische Constraints, die *semantische* bzw. *pragmatische* und *topologische* bzw. *geometrische* sowie *temporäre* Relationen (bei animierten Darstellungen) spezifizieren. Sie eignen sich insbesondere auch zur Repräsentation lokaler Randbedingungen,

wie von außen vorgegebene Formatrestriktionen, Randbereiche, Abstände, Überlappungsfreiheit etc.

Die Struktur eines multimodalen Dokuments wird im wesentlichen durch funktionale Abhängigkeiten zwischen den einzelnen Dokumentteilen, also Bildern, Texten oder Bild-Textkombinationen, die ihrerseits wieder Aktionen repräsentieren können, bestimmt. Die inhaltlichen Beziehungen zwischen Bildern und Texten werden vom Präsentationsplaner in Form von sog. *'rhetorischen Relationen'*, basierend auf der von Mann und Thompson vorgeschlagenen *RST-Theorie* (vgl. [23]) zum Aufbau von Texten, spezifiziert. Solche RST-Relationen sind z.B. *'sequence'*, *'contrast'*, *'elaboration'*, *'organization'*, *'motivation'*, etc. Für einige dieser semantisch-pragmatischen Kohärenzrelationen sind geeignete Designheuristiken bekannt, die sie in Form graphischer Constraints adäquat reflektieren (vgl. auch [17, 19]).

Über diese rein inhaltliche Klassifizierung der Constraints hinaus ergibt sich eine Priorisierung dieser nach deren Einfluß auf die Gesamtpräsentation, in *obligatorische* Constraints, deren Erfülltsein für die Verständlichkeit des Dokumentes unabdingbar ist (z.B. Überlappungsfreiheit), *optionale* Constraints, die über eine Präferenzskala geordnet sind und ästhetisch wünschenswerte Objektkonstellationen spezifizieren (z.B. optimale Distanz zwischen zwei Boxen) und *Default-Constraints*. Letztere repräsentieren a priori Annahmen über Dokumenttypen, die solange gelten, bis sie von neuen stärkeren Constraints überschrieben werden. Durch die Einführung einer Präferenzskala über dem Constraint-Netz wird somit eine Hierarchisierung der Constraints möglich. Das hier verwendete Constraint-Verfahren orientiert sich an einem von Borning et al. (vgl. [7]) vorgeschlagenen Ansatz zur Hierarchisierung von Constraints.

Ein typisches Beispiel für die Benutzung einer Constraint-Hierarchie im geometrischen Layout ist das Problem der Bestimmung des Leerraums zwischen zwei kontrastierenden Graphiken $G1$ $G2$. Die entsprechenden Designkriterien lassen sich durch folgende zwei Constraints repräsentieren:

- Ein obligatorisches Constraint spezifiziert, daß der Abstand zwischen $G1$ und $G2 > 0$ sein muß.
- Eine Disjunktion von zwei optionalen Constraints gibt an, daß die Graphiken bevorzugt horizontal und ansonsten vertikal zu alignieren sind.

Neben Constraints, die statische Designheuristiken repräsentieren, müssen dynamische Constraints *on the fly* aus den vom Präsentationsplaner spezifizierten inhaltlichen Relationen kompiliert werden. Zu den dynamisch zu generierenden Constraints zählen auch solche, die sich auf Relationen zwischen einer a priori nicht bekannten Anzahl von Objekten beziehen (vgl. z.B. *'sequence'*).

Da bei Syntheseaufgaben, wie z.B. Konfigurationsaufgaben, viele Constraints nur lokalen Einfluß haben, wird die für die Lösung relevante Variablenmenge und damit das Constraint-Netz bzw. die Constraint-Hierarchie häufig dynamisch in Abhängigkeit des Problemlösungsprozesses verändert (vgl. auch [26]). Wir verwenden deshalb einen inkrementellen Constraint-Propagierungsalgorithmus, der es erlaubt Constraints zur Laufzeit hinzuzufügen bzw. zu löschen (vgl. Kap. 5).

Der vorgestellte Constraint-Formalismus erlaubt die Aggregation von sog. primitiven Constraints, die auf arithmetischen Ausdrücken arbeiten und elementare geometrische Beziehungen (wie z.B. 'beside', 'under') repräsentieren, zu zusammengesetzten Constraints (*Compound-Constraints*). Ein Beispiel für ein Compound-Constraint ist die Repräsentation des graphischen Constraints für die Relation 'contrast'. Abb. 2 zeigt eine Definition dieses Constraints in prädikatenlogischer Notation, in der zwei verschiedene Designalternativen zur Realisierung der Kontrastrelation spezifiziert sind. In Abb. 3 wird diese Definition durch eine entsprechende Netzwerkdarstellung illustriert.

$$\begin{aligned}
 \text{CONTRAST } (G_1, G_2) \leftrightarrow & \\
 G_1 \equiv \text{pkt}(x_{G_1}, y_{G_1}, wi_{G_1}, he_{G_1}) \wedge & \\
 G_2 \equiv \text{pkt}(x_{G_2}, y_{G_2}, wi_{G_2}, he_{G_2}) \wedge & \\
 [\text{EQUAL}(y_{G_1}, y_{G_2}) \wedge \text{BESIDE}(x_{G_1}, wi_{G_1}, x_{G_2}) & \\
 \vee & \\
 \text{EQUAL}(x_{G_1}, x_{G_2}) \wedge \text{UNDER}(y_{G_1}, he_{G_1}, y_{G_2})] &
 \end{aligned}$$

Abbildung 2: Repräsentation des Compound-Constraints *CONTRAST*

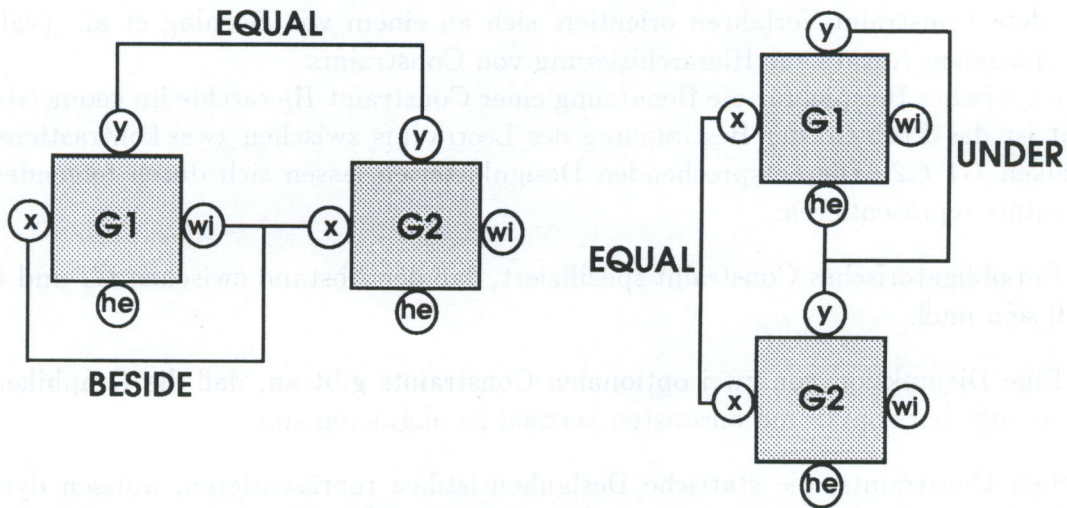


Abbildung 3: Netzwerkdarstellung der Definition aus Abb. 2

5 Ein mehrschichtiger Constraint-Solver zur Verarbeitung graphischen Wissens

Die Planung des Layoutprozesses läuft in zwei Phasen auf drei unterschiedlich detaillierten Ebenen ab. In der ersten Phase wird ein Groblayout für die vom Planer spezifizierten unin-

stanziierten Text- und Graphikboxen bestimmt. Dabei benutzt die Layout-Komponente Defaultannahmen u.a. über Designstrategien zum Arrangement der Layoutobjekte im zweidimensionalen Dokumentraum.

Nach Abschluß der Generierungsprozesse werden die bisher leeren Boxen mit den von den media-spezifischen Generatoren erzeugten aktuellen Graphiken bzw. Texten instanziiert. Daraufhin kann die zweite Phase das Feinlayout planen, wobei den Text- und Graphikelementen reale Koordinaten im Layoutraum zugewiesen werden.

Das funktionale Layout des Dokumentes wird durch ein Grid unterstützt, was aus den geometrischen Grunddaten der Objekte, den Generierungsparametern und dem dokumentenspezifischen Wissen dynamisch bestimmt wird (vgl. [20]). Die Layoutobjekte werden entsprechend dem Grid so auf der zweidimensionalen Plazierungsebene angeordnet, daß ein Objekt eine ganzzahlige Menge von Rasterfeldern überdeckt.

Die für die Layouterstellung relevanten Beziehungen lassen sich in lokale und globale Beziehungen unterteilen. Lokale Beziehungen sind inhaltliche Relationen zwischen Layoutobjekten, die vom Planer vorgegeben werden. Sie fassen Objekte teilweise verschiedener Modalität zu größeren Einheiten zusammen. Globale Beziehungen sind Abbildungen geometrischer Relationen, die Einheiten zueinander plazieren. Diese können vom Planer explizit vorgegeben werden oder werden mittels einer Heuristik geeignet zu der jeweiligen Einheit ausgewählt. Neben inhaltlichen Daten, wie etwa der Typ eines Objektes, sind die geometrischen Ausmaße bei der Plazierung von grundlegender Bedeutung. Hierzu wird ein Objekt durch den kleinsten ihn umschließenden Rahmen modelliert.

Eine zentrale Idee dieses Ansatzes ist die Unterteilung des Plazierungsprozesses in die Bestimmung der lokalen Topologie und der hieraus folgenden expliziten Berechnung der geometrischen Plazierung im Grid. Diese Aufgabe wird von zwei dedizierten inkrementellen Constraint-Solvern übernommen, die in einem hierarchischen Modell organisiert sind und über Constraint-Hierarchien bzw. nach dem Domänen-Konzept arbeiten (s. Abb. 4).

Um ein effizientes Verhalten eines solchen mehrschichtigen Constraint-Systems zu garantieren, ist eine geeignete Ansteuerung der verschiedenen Kontrollmechanismen erforderlich. Das geschieht auf der obersten Ebene des Solver-Modells, die Metawissen in Form von Designheuristiken über die Abarbeitung der vom Präsentationsplaner spezifizierten Dokumentrepräsentation enthält und so den Evaluierungsprozeß steuert. Dieses Metawissen ist zum Teil deklarativ über Default-Constraints als auch durch Regeln repräsentiert.

5.1 Verarbeitung lokaler Beziehungen

Da das Layout durch eine klare Aufteilung geprägt wird, die z.B. Zusammenhänge und Gegensätze in der zu präsentierenden Information reflektiert, wird dies in der Designstrategie durch die Bildung lokaler Einheiten berücksichtigt. Lokale Beziehungen ergeben sich aus den vom Planer vorgegebenen semantisch-pragmatischen Relationen zwischen den Objekten und betreffen die relative Lage der Objekte untereinander. Die entsprechenden Layout-Constraints repräsentieren topologische Anordnungsmöglichkeiten, die durch die Relationen impliziert werden (s. Kap. 4).

Durch die feste Vorgabe der Beziehungen zwischen den Layoutobjekten bietet sich ein Constraint-Solver an, der nach der *Value-Inference*-Technik arbeitet und feste Werte durch

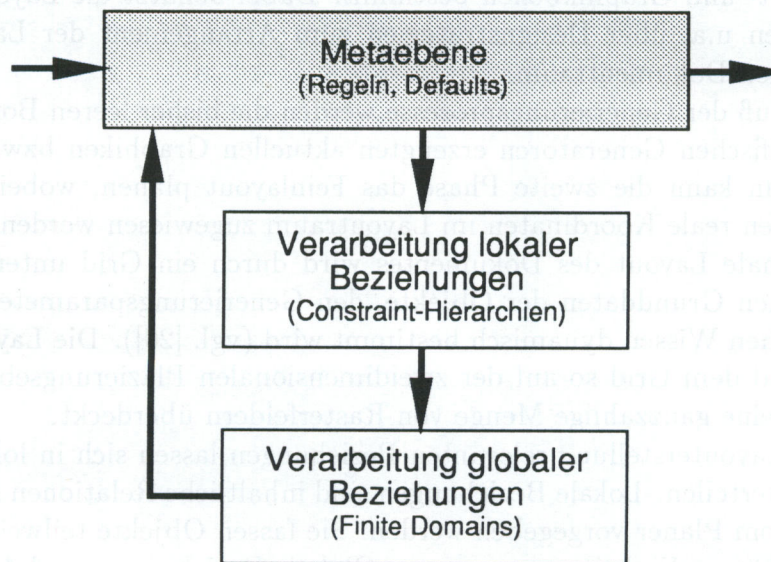


Abbildung 4: Modell des mehrschichtigen Constraint-Solvers

das Constraint-Netz propagiert (vgl. auch [36, 35, 6]). Damit werden die Objekte relativ zum Koordinatenursprung positioniert und erst später durch die Anwendung globaler Beziehungen an explizite Koordinaten gebunden.

Da man insbesondere bei der Generierung multimodaler Präsentationen wie z.B. in WIP Constraints häufig nach ihrer Stärke gewichten will, haben wir einen speziellen Constraint-Solver, ähnlich dem von Freeman-Benson et al. entwickelten *DeltaBlue*-Algorithmus (vgl. [13]) verwendet, der die Bearbeitung von Constraint-Hierarchien unterstützt.

Die *Inkrementalität* dieses Solvers erlaubt es, zur Laufzeit Änderungen am Constraint-Netz vorzunehmen, ohne das gesamte Netz neuzuberechnen, d.h. es können dynamisch Constraints hinzugefügt bzw. gelöscht werden. Aus allen zulässigen Lösungen zu einer Constraint-Hierarchie berechnet der inkrementelle Constraint-Solver unter Verwendung eines schnellen lokalen Propagierungsalgorithmus mittels des sog. *local-predicate-better* Komparators (vgl. [13]) die beste Lösung.

Da die hier verwendete Constraint-Sprache über die Möglichkeit zur Definition von Compound-Constraints verfügt, lassen sich durch die Angabe von alternativen Basis-Constraints verschiedene graphische Darstellungsformen für die lokalen Beziehungen ausdrücken, wobei der Constraint-Solver entscheidet, in welcher Reihenfolge die einzelnen Basis-Constraints evaluiert werden. Können die Constraints einer Designstrategie nicht erfüllt werden, so wird durch *Backtracking* eine andere ausgewählt.

5.2 Verarbeitung globaler Beziehungen

Im Gegensatz zu den lokalen Beziehungen sind die globalen Beziehungen schwächerer Natur. Sie verbinden Einheiten von lokal zusammenhängenden Objekten entsprechend der vorliegenden Dokumentstruktur und arrangieren diese nach dokumentenspezifischen Designheuristiken relativ zu anderen Einheiten im Layoutraum. Durch die Verwendung eines

Design-Grids erhalten wir eine zusätzliche Einschränkung des Layoutraums. Diese läßt sich ebenfalls über Constraints repräsentieren, wobei dann die Grid-Koordinaten endliche Wertebereiche (*Finite-Domains*) für die Variablen beschreiben.

Ähnlich dem Repräsentationsformalismus lokaler Beziehungen können wir Constraint-Netze verwenden, um globale Beziehungen zu repräsentieren. Da für jede Einheit, nach Einbettung in das bestehende Constraint-Netz, im allgemeinen nur eine endliche Anzahl von möglichen Plazierungsflächen zur Auswahl steht, verwenden wir einen speziellen Constraint-Ansatz, der auf endlichen, diskretwertigen Intervallen von natürlichen Zahlen arbeitet.

Die Ideen bei der Entwicklung des Constraint-Solvers zur Behandlung von Finite-Domains gehen auf die Arbeit von Van Hentenryck (vgl. [37]) im Rahmen des *CHIP*-Projektes zum *Constraint Logic Programming* am ECRC zurück. Der Propagierungsalgorithmus benutzt hier ähnlich der *Label-Inference*-Technik (vgl. [22, 9]) Constraints zur Einschränkung der Wertemengen. Als effizienten Inferenzmechanismus verwenden wir zur *a priori* Einschränkung des Suchraumes eine Prozedur ähnlich dem *Forward-Checking* in *CHIP* (vgl. [37]). Dadurch wird frühzeitig eine Nichterfüllbarkeit der Relationenmenge erkannt, so daß von der Kontrollebene die nächste Alternative ausgewählt werden kann. Der Constraint-Solver arbeitet inkrementell, d.h. bei Einführung neuer Constraints wird nur ein Teil des Netzes neu berechnet.

Durch die Angabe von neuen unteren und oberen Grenzen für die Einheiten werden die Wertebereiche der Constraint-Variablen weiter restringiert, bis diese nur noch aus einem Element bestehen. Um in der Plazierung möglichst flexibel zu sein, erhalten globale Beziehungen die maximale Plazierungsfreiheit in X- und Y-Richtung ohne dadurch ein Constraint zu verletzen. Enthält ein Intervall nur noch einen Wert, so ist die Einheit in dieser Variablen festgelegt. Trifft dies sowohl für die X- als auch für die Y-Richtung zu, so vermerkt ein internes *Gridgedächtnis* die Belegung und blockiert den entsprechenden Teil der Plazierungsfläche. Auf diese Weise wird eine global-konsistente Plazierung garantiert, in der sich keine Einheiten überlappen können.

6 Resümee

Die automatische Layoutgestaltung zählt zur großen Menge diskreter Kombinatorikprobleme, die als NP-vollständig nachgewiesen sind. In der vorliegenden Arbeit wurde gezeigt, daß sich das Layoutproblem als CSP auffassen läßt und durch die Verwendung von Such- und Konsistenztechniken zur Beschränkung des Layoutraums effektiv lösbar ist. Das Ziel dieser Arbeit bestand darin, am Beispiel des Layout-Managers in *WIP*, Möglichkeiten zur Repräsentation und Verarbeitung von design-relevantem graphischen Wissen bei der automatischen Generierung des Layouts von multimodalen Präsentationen aufzuzeigen. Es wurde dazu ein neues hierarchisches Verarbeitungsmodell, bestehend aus zwei speziellen inkrementellen Constraint-Solvern, zur Bearbeitung von Constraint-Hierarchien bzw. Finite-Domains, die von einer Metaebene aus regel-basiert angesteuert werden, vorgestellt. Die Effizienz dieses Constraint-Verfahrens zeigt sich in ersten Performanzanalysen, diese ergaben für das Layout einer Dokumentseite bestehend aus

acht Graphik-Text-Blöcken Laufzeiten unter fünf Sekunden. Ein Prototyp des Layout-Managers (vgl. [20]) wurde auf Apple MacIvory in Common Lisp/CLOS implementiert und in das SystemWIP integriert.

Literatur

- [1] E. André and T. Rist. Synthesizing illustrated documents: A plan-based approach. In *Proceedings of InfoJapan '90, Vol. 2*, pages 163–170, Tokyo, 1990. Also DFKI Research Report RR-91-06.
- [2] Y. Arens, E. Hovy, and M. Vossers. Describing the knowledge underlying the processing of multimedia instruction manuals. Submitted to *Cognitive Science '91*, 1991.
- [3] R. Arnheim, editor. *Visual Thinking*. Faber and Faber, London, 1969.
- [4] N. Badler and B. Webber. Task communication through natural language and graphics: A workshop report. Dept. of Computer and Information Science, University of Pennsylvania, June 1990.
- [5] R. Beach. *Setting Tables and Illustrations with Style*. PhD thesis, Dept. of Computer Science, University of Waterloo, Ontario, 1985.
- [6] A. Borning. The programming language aspects of ThingLab, a constraint-oriented simulation laboratory. *ACM Transactions on Programming Languages and Systems*, 3(4):353–387, October 1981.
- [7] A. Borning, R. Duisberg, B. Freeman-Benson, A. Kramer, and M. Woolf. Constraint hierarchies. In *Proceedings of OOPSLA '87*, pages 48–60, October 1987.
- [8] G. Braun, editor. *Grundlagen der visuellen Kommunikation*. Bruckmann, München, 1987.
- [9] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32:281–331, 1987.
- [10] S. Feiner. APEX: An experiment in the automated creation of pictorial explanations. *IEEE Computer Graphics and Applications*, 5(11):29–39, 1985.
- [11] S. Feiner. A grid-based approach to automating display layout. In *Proceedings of the Graphics Interface '88*. Morgan Kaufmann, Los Altos, CA, June 1988.
- [12] S. Feiner. An architecture for knowledge-based graphical interfaces. In Sullivan and Tyler [34].
- [13] B. Freeman-Benson, J. Maloney, and A. Borning. An incremental constraint solver. *Communications of the ACM*, 33(1):54–63, 1990.

- [14] M. Friedell. Automatic synthesis of graphical object descriptions. *Computer Graphics*, 18(3):53–62, 1984.
- [15] J. Gosling. *Algebraic Constraints*. PhD thesis, Dept. of Computer Science, Carnegie Mellon University, 1983.
- [16] W. Graf. Spezielle Aspekte des automatischen Layout-Designs bei der koordinierten Generierung von multimodalen Dokumenten. GI-Workshop 'Multimediale elektronische Dokumente', Heidelberg, November 1990.
- [17] W. Graf. Constraint-based processing of design knowledge. In *Proceedings of the AAAI-91 Workshop on Intelligent Multimedia Interfaces*, Anaheim, CA, July 1991.
- [18] W. Leler, editor. *Constraint Programming Languages: Their Specification and Generation*. Addison-Wesley, Reading, MA, 1988.
- [19] W. Maaß. Constraint-basierte Repräsentation von graphischem Wissen am Beispiel des Layout-Managers in WIP. Master's thesis, Dept. of Computer Science, University of Saarbrücken, 1991. forthcoming.
- [20] W. Maaß, T. Schiffmann, and D. Soetopo. LAYLAB: Ein System zur automatischen Platzierung in multimodalen Dokumenten. Fortgeschrittenenpraktikum 'Wissensbasierte Graphikgenerierung', Dept. of Computer Science, University of Saarbrücken, 1991.
- [21] J. Mackinlay. *Automatic Design of Graphical Presentations*. PhD thesis, Dept. of Computer Science, Stanford University, Stanford, CA, 1985.
- [22] A. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977.
- [23] W. Mann and S. Thompson. Rhetorical structure theory: Towards a functional theory of text organization. *TEXT*, 8(3), 1988.
- [24] J. Marks. The competence of an automated graphic designer. In *Proceedings of the 1991 Long Island Conference on Artificial Intelligence and Computer Graphics*, NYIT, pages 53–61, New York, March 1991.
- [25] J. Marks and E. Reiter. Avoiding unwanted conversational implicatures in text and graphics. In *Proceedings of the 8th National Conference of the American Association for Artificial Intelligence*, pages 450–456, Boston, MA, July 1990.
- [26] S. Mittal and B. Falkenhainer. Dynamic constraint satisfaction problems. In *Proceedings of the 8th National Conference of the American Association for Artificial Intelligence*, pages 25–32, Boston, MA, July 1990.
- [27] U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Science*, 7(2):95–132, 1974.

- [28] J. Müller-Brockmann, editor. *Grid Systems in Graphic Design*. Verlag Arthur Niggli, Niederteufen, Switzerland, 1981.
- [29] D. Navinchandra, editor. *Exploration and Innovation in Design*. Springer-Verlag, Berlin, Germany, 1991.
- [30] G. Nelson. Juno, a constraint-based graphics system. *Proceedings of the SIGGRAPH '85*, 19(3):235–243, 1985.
- [31] S. Roth, J. Mattis, and X. Mesnard. Graphics and natural language as components of automatic explanation. In Sullivan and Tyler [34].
- [32] D. Seligmann and S. Feiner. Specifying composite illustrations with communicative goals. In *Proceedings of the UIST '89 (ACM SIGGRAPH Symp. on User Interface Software and Technology)*, pages 1–9, Williamsburg, VA, November 1988.
- [33] A. Stankowski and K. Duschek, editors. *Visuelle Kommunikation*. Dietrich Reimer, Berlin, 1989.
- [34] J. Sullivan and S. Tyler, editors. *Architectures for Intelligent User Interfaces: Elements and Prototypes*. Addison-Wesley, Reading, MA, 1990. Proc. of the Workshop on Intelligent Interfaces, Monterey, CA, March 29-April 1, 1988.
- [35] G. Sussman and G. Steele. Constraints – a language for expressing almost-hierarchical descriptions. *Artificial Intelligence*, 14(1):1–39, 1980.
- [36] I. Sutherland. Sktechpad: A man-machine graphical communication system. In *IFIPS Proceedings of the Spring Joint Computer Conference*, pages 329–345, 1963.
- [37] P. van Hentenryck, editor. *Constraint Satisfaction in Logic Programming*. MIT Press, Cambridge, MA, 1989. Revision of Ph.D. thesis, University of Namur, 1987.
- [38] C. van Wyk. A high-level language for specifying pictures. *ACM Transactions on Graphics*, 1(2):163–182, 1982.
- [39] W. Wahlster, E. André, S. Bandyopadhyay, W. Graf, and T. Rist. WIP: The coordinated generation of multimodal presentations from a common representation. In *Computational Theories of Communication and their Applications*. Springer-Verlag, Berlin, Germany, 1991. Also DFKI Research Report RR-91-08.
- [40] W. Wahlster, E. André, W. Graf, and T. Rist. Designing illustrated texts: How language production is influenced by graphics generation. In *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics*, pages 8–14. Springer-Verlag, Berlin, Germany, April 1991. Also DFKI Research Report RR-91-05.
- [41] D. Willows and H. Houghton, editors. *The Psychology of Illustration, Vol. 1, 2*. Springer-Verlag, Berlin, Germany, 1987.



**Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH**

DFKI
-Bibliothek-
PF 2080
6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen oder die aktuelle Liste von erhältlichen Publikationen können bezogen werden von der oben angegebenen Adresse.

Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of currently available publications can be ordered from the above address.

The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-90-03

Andreas Dengel, Nelson M. Mattos: Integration of Document Representation, Processing and Management
18 pages

RR-90-04

Bernhard Hollunder, Werner Nutt: Subsumption Algorithms for Concept Languages
34 pages

RR-90-05

Franz Baader: A Formal Definition for the Expressive Power of Knowledge Representation Languages
22 pages

RR-90-06

Bernhard Hollunder: Hybrid Inferences in KL-ONE-based Knowledge Representation Systems
21 pages

RR-90-07

*Elisabeth André, Thomas Rist: Wissensbasierte Informationspräsentation:
Zwei Beiträge zum Fachgespräch Graphik und KI:*

1. Ein planbasierter Ansatz zur Synthese illustrierter Dokumente
2. Wissensbasierte Perspektivenwahl für die automatische Erzeugung von 3D-Objektdarstellungen

24 Seiten

RR-90-08

Andreas Dengel: A Step Towards Understanding Paper Documents
25 pages

RR-90-09

Susanne Biundo: Plan Generation Using a Method of Deductive Program Synthesis
17 pages

RR-90-10

Franz Baader, Hans-Jürgen Bürckert, Bernhard Hollunder, Werner Nutt, Jörg H. Siekmann: Concept Logics
26 pages

RR-90-11

Elisabeth André, Thomas Rist: Towards a Plan-Based Synthesis of Illustrated Documents
14 pages

RR-90-12

Harold Boley: Declarative Operations on Nets
43 pages

RR-90-13

Franz Baader: Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles
40 pages

RR-90-14

Franz Schmalhofer, Otto Kühn, Gabriele Schmidt: Integrated Knowledge Acquisition from Text, Previously Solved Cases, and Expert Memories
20 pages

RR-90-15

Harald Trost: The Application of Two-level Morphology to Non-concatenative German Morphology
13 pages

RR-90-16

Franz Baader, Werner Nutt: Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification
25 pages

RR-90-17

Stephan Busemann: Generalisierte Phasenstrukturgrammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung
114 Seiten

RR-91-01

Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, Gert Smolka: On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations
20 pages

RR-91-02

Francesco Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, Werner Nutt: The Complexity of Existential Quantification in Concept Languages
22 pages

RR-91-03

B.Hollunder, Franz Baader: Qualifying Number Restrictions in Concept Languages
34 pages

RR-91-04

Harald Trost: X2MORF: A Morphological Component Based on Augmented Two-Level Morphology
19 pages

RR-91-05

Wolfgang Wahlster, Elisabeth André, Winfried Graf, Thomas Rist: Designing Illustrated Texts: How Language Production is Influenced by Graphics Generation.
17 pages

RR-91-06

Elisabeth André, Thomas Rist: Synthesizing Illustrated Documents A Plan-Based Approach
11 pages

RR-91-07

Günter Neumann, Wolfgang Finkler: A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures
13 pages

RR-91-08

Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist: WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation
23 pages

RR-91-09

Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds
31 pages

RR-91-10

Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages
31 pages

RR-91-11

Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches
37 pages

RR-91-12

J.Mark Gawron, John Nerbonne, Stanley Peters: The Absorption Principle and E-Type Anaphora
33 pages

RR-91-13

Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming
17 pages

RR-91-14

Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I
27 pages

RR-91-15

Bernhard Nebel, Gert Smolka: Attributive Description Formalisms ... and the Rest of the World
20 pages

RR-91-16

Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations
18 pages

RR-91-17

Andreas Dengel, Nelson M. Mattos: The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18

John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein: A Diagnostic Tool for German Syntax
20 pages

RR-91-19

Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20

Christoph Klauck, Ansgar Bernardi, Ralf Legleitner: FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21

Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22

Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23

Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24

Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25

Karin Harbusch, Wolfgang Finkler, Anne Schauder: Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26

M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger: Integrated Plan Generation and Recognition - A Logic-Based Approach -
17 pages

RR-91-27

A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer: ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28

Rolf Backofen, Harald Trost, Hans Uszkoreit: Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-29

Hans Uszkoreit: Strategies for Adding Control Information to Declarative Grammars
17 pages

RR-91-30

Dan Flickinger, John Nerbonne: Inheritance and Complementation: A Case Study of Easy Adjectives and Related Nouns
39pages

RR-91-31

H.-U. Krieger, J. Nerbonne: Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32

Rolf Backofen, Lutz Euler, Günther Görz: Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

RR-91-35

Winfried Graf, Wolfgang Maaß: Constraint-basierte Verarbeitung graphischen Wissens
14 Seiten

DFKI Technical Memos
TM-90-03

Franz Baader, Bernhard Hollunder: KRIS: Knowledge Representation and Inference System -System Description-
15 pages

TM-90-04

Franz Baader, Hans-Jürgen Bürckert, Jochen Heinsohn, Bernhard Hollunder, Jürgen Müller, Bernhard Nebel, Werner Nutt, Hans-Jürgen Profitlich: Terminological Knowledge Representation: A Proposal for a Terminological Logic
7 pages

TM-91-01

Jana Köhler: Approaches to the Reuse of Plan Schemata in Planning Formalisms
52 pages

TM-91-02

Knut Hinkelmann: Bidirectional Reasoning of Horn Clause Programs: Transformation and Compilation
20 pages

TM-91-03

Otto Kühn, Marc Linster, Gabriele Schmidt: Clamping, COKAM, KADS, and OMOS: The Construction and Operationalization of a KADS Conceptual Model
20 pages

TM-91-04

Harold Boley (Ed.): A sampler of Relational/Functional Definitions
12 pages

TM-91-05

Jay C. Weber, Andreas Dengel, Rainer Bleisinger: Theoretical Consideration of Goal Recognition Aspects for Understanding Information in Business Letters
10 pages

TM-91-06

Johannes Stein: Aspects of Cooperating Agents
22 pages

TM-91-08

Munindar P. Singh: Social and Psychological Commitments in Multiagent Systems
11 pages

TM-91-09

Munindar P. Singh: On the Semantics of Protocols Among Distributed Intelligent Agents
18 pages

TM-91-10

Béla Buschauer, Peter Poller, Anne Schauder, Karin Harbusch: Tree Adjoining Grammars mit Unifikation
149 pages

TM-91-11

Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

TM-91-12

Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13

Knut Hinkelmann: Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

DFKI Documents**D-90-05**

Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Formalismus zur Repräsentation von Geo-metrie- und Technologieinformationen als Teil eines Wissensbasierten Produktmodells
66 Seiten

D-90-06

Andreas Becker: The Window Tool Kit
66 Seiten

D-91-01

Werner Stein, Michael Sintek: Relfun/X - An Experimental Prolog Implementation of Relfun
48 pages

D-91-03

Harold Boley, Klaus Elsbernd, Hans-Günther Hein, Thomas Krause: RFM Manual: Compiling RELFUN into the Relational/Functional Machine
43 pages

D-91-04

DFKI Wissenschaftlich-Technischer Jahresbericht 1990
93 Seiten

D-91-06

Gerd Kamp: Entwurf, vergleichende Beschreibung und Integration eines Arbeitsplanerstellungssystems für Drehteile
130 Seiten

D-91-07

Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: TEC-REP: Repräsentation von Geometrie- und Technologieinformationen
70 Seiten

D-91-08

Thomas Krause: Globale Datenflußanalyse und horizontale Compilation der relational-funktionalen Sprache RELFUN
137 pages

D-91-09

David Powers and Lary Reeker (Eds.): Proceedings MLNLO '91 - Machine Learning of Natural Language and Ontology
211 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-\$).

D-91-10

Donald R. Steiner, Jürgen Müller (Eds.): MAAMAW '91: Pre-Proceedings of the 3rd European Workshop on „Modeling Autonomous Agents and Multi-Agent Worlds“
246 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-\$).

D-91-11

Thilo C. Horstmann: Distributed Truth Maintenance
61 pages

D-91-12

Bernd Bachmann: Hiera_{Con} - a Knowledge Representation System with Typed Hierarchies and Constraints
75 pages

D-91-13

International Workshop on Terminological Logics Organizers: Bernhard Nebel, Christof Peltason, Kai von Luck
131 pages

D-91-14

Erich Achilles, Bernhard Hollunder, Armin Laux, Jörg-Peter Mohren: KRIS: Knowledge Representation and Inference System - Benutzerhandbuch -
28 Seiten

