



Deutsches  
Forschungszentrum  
für Künstliche  
Intelligenz GmbH

**Document**

D-97-05

**Anforderungen an ein System zur  
Dokumentanalyse im Unternehmenskontext**

—  
**Integration von Datenbeständen, Aufbau- und  
Ablauforganisation**

**Stephan Baumann, Majdi Ben Hadj Ali, Jürgen Lichter,  
Michael Malburg, Harald Meyer auf'm Hofe, Claudia Wenzel**

**September 1997**

**Deutsches Forschungszentrum für Künstliche Intelligenz  
GmbH**

Postfach 20 80  
67608 Kaiserslautern, FRG  
Tel.: + 49 (631) 205-3211  
Fax: + 49 (631) 205-3210

Stuhlsatzenhausweg 3  
66123 Saarbrücken, 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 and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, Sema Group, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry of Education, Science, 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
- Computer Linguistics
- Programming Systems
- Deduction and Multiagent Systems
- Document Analysis and Office Automation.

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.

Dr. Dr. D. Ruland  
Director

# **Anforderungen an ein System zur Dokumentanalyse im Unternehmenskontext — Integration von Datenbeständen, Aufbau- und Ablauforganisation**

**Stephan Baumann, Majdi Ben Hadj Ali, Jürgen Lichter, Michael Malburg, Harald Meyer auf'm Hofe, Claudia Wenzel**

DFKI-D-97-05

This work has been supported by a grant from The Federal Ministry of Education, Science, Research, and Technology (FKZ ITWM-9702).

© Deutsches Forschungszentrum für Künstliche Intelligenz 1997

This work may not be copied or reproduced in whole or part for any commercial purpose. Permission to copy in whole or 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 the Deutsche 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.

ISSN 0946-0098

Anforderungen an ein System zur  
Dokumentanalyse im Unternehmenskontext —  
Integration von Datenbeständen, Aufbau- und  
Ablauforganisation

Stephan Baumann, Majdi Ben Hadj Ali, Jürgen Lichter,  
Michael Malburg, Harald Meyer auf'm Hofe, Claudia Wenzel

September 1997

## **Zusammenfassung**

Workflowmanagementsysteme werden im Bürobereich verstärkt zur effizienten Geschäftsprozeßabwicklung eingesetzt. Das bereits Mitte der 70er Jahre propagierte papierlose Büro bleibt jedoch gegenwärtig immer noch Utopie, da auch durch den allgegenwärtigen Einsatz von Computern im Bürobereich der Durchsatz an Schriftstücken nicht gesenkt wird.

Insbesondere die Handhabung von papierintensiven Vorgängen ist in hohem Maße abhängig von einer Identifikation und Aufbereitung der in den Dokumenten enthaltenen Informationen. Allerdings müssen solche Daten z. B. bei eingehender Post immer noch von Hand eingegeben werden.

In diesem Dokument werden Anforderungen an ein System aufgestellt, das diesen Medienbruch überwinden soll. Techniken aus dem Gebiet der Dokumentanalyse und des Dokumentverstehens werden in den Workflowkontext integriert und nutzen das dort verfügbare Wissen zur Steigerung der Erkennungsqualität. Durch Einschränkung des aktuellen Kontextes – etwa in Form offener Vorgänge – soll eine Erhöhung der Erkennungspräzision erreicht werden.

Bei der Beschreibung der Systemanforderungen wurde nach den Richtlinien des V-Modells vorgegangen.



# Inhaltsverzeichnis

1	Allgemeines .....	1
1.1	Übersicht .....	1
1.2	Abkürzungen.....	1
1.3	Definitionen.....	1
1.4	Referenzierte Dokumente .....	3
2	Zweck des Dokuments .....	5
3	Analyseergebnisse.....	7
3.1	Ist-Aufnahme/-Analyse.....	7
3.1.1	Ist-Aufnahme .....	7
3.1.2	Ist-Analyse .....	7
3.2	Bedrohungs- und Risikoanalyse.....	7
4	Allgemeine Anforderungen an das System.....	9
4.1	Gesamtfunktion des Systems .....	9
4.2	Anforderungen hinsichtlich der Benutzung/Nutzung des Systems.....	9
4.3	Anforderungen hinsichtlich der Realisierung interner Schnittstellen .	10
4.4	Kritikalität des Systems .....	11
4.5	DV-Sicherheit.....	11
4.6	Anforderungen an sonstige Qualitätsmerkmale des Systems .....	11
5	Anforderungen an die Entwicklungs- und SWPÄ-Umgebung .....	13
6	Anforderungen an die systemexternen Schnittstellen .....	15
6.1	Einsatzumgebung .....	15
6.1.1	Technische Einsatzumgebung .....	15
6.1.2	Organisatorische Einbettung .....	16
6.1.3	Nutzung.....	16
6.2	Sonstige systemexterne Schnittstellen .....	16
6.2.1	Übersicht .....	16
6.2.2	Einzelbeschreibungen .....	17
6.2.2.1	Schnittstelle Workflow-Beschreibung/-Design.....	17
6.2.2.2	Schnittstelle Workflow-Anreicherung .....	17
6.2.2.3	Schnittstelle Workflow-Aktionen.....	17
6.2.2.4	Schnittstelle Dokumentverifikation .....	17
6.2.2.5	Schnittstelle On-Line-Teaching .....	18
6.2.2.6	Schnittstelle Off-Line-Learning.....	18
6.2.2.7	Schnittstelle Systemkonfiguration .....	18
6.2.2.8	Schnittstelle Testumgebung .....	19
6.2.2.9	Schnittstelle Dokumente .....	20
6.2.2.10	Schnittstelle Monitoring und Tuning.....	20

7	Funktions- und datenorientierte fachliche Anforderungen an das System aus objektorientierter Sicht .....	21
7.1	Anwendungsszenarien .....	21
7.1.1	Off-Line Lernphase für einzelne Spezialisten.....	21
7.1.2	Konfigurationsphase.....	21
7.1.3	Analysephase .....	22
7.1.4	Verifikation analysierter Dokumente .....	22
7.1.5	Aktualisieren von Workflow- und Dokumentdaten.....	24
7.2	Klassenkategorien und Rollen des Problembereichs .....	25
7.3	Klassenkategorien- und Rollenbeschreibungen .....	26
7.3.1	Workflow-Engine .....	26
7.3.1.1	Methode: StartEngine .....	26
7.3.1.2	Methode: StopEngine .....	27
7.3.2	Workflow-Spezifikation .....	27
7.3.2.1	Methode: NewWorkflow .....	27
7.3.2.2	Methode: ChangeWorkflow .....	28
7.3.2.3	Methode: EnrichWorkflow.....	28
7.3.2.4	Methode: DeleteWorkflow .....	29
7.3.3	Workflow-Applikation .....	29
7.3.3.1	Methode: GetOpenProcesses .....	29
7.3.3.2	Methode: WorkProcess .....	30
7.3.3.3	Methode: GetStateOfProcess.....	30
7.3.3.4	Methode: SetSubstitution.....	30
7.3.3.5	Methode: SendMail.....	30
7.3.4	Konfigurationskomponente .....	31
7.3.4.1	Methode: ViewConcepts .....	32
7.3.4.2	Methode: AddConcept .....	32
7.3.4.3	Methode: AddConstraint.....	32
7.3.4.4	Methode: AddConfigurationManually.....	33
7.3.4.5	Methode: AddConfigurationAutomatically .....	33
7.3.4.6	Methode: AddSubsumptionRelation.....	33
7.3.4.7	Methode: AddPartWholeRelation.....	34
7.3.4.8	Methode: ModifyAttribute.....	34
7.3.5	DAU-Monitor .....	34
7.3.5.1	Methode: ViewSystemConfiguration.....	34
7.3.5.2	Methode: ChangeSystemConfiguration .....	35
7.3.6	Verifikation und Akquisition .....	35
7.3.6.1	Methode: VerifyResults .....	36
7.3.6.2	Methode: CheckAll.....	36
7.3.6.3	Methode: AcquireLexicalKnowledge .....	36
7.3.6.4	Methode: AcquirePatterns .....	37
7.3.6.5	Methode: AcquireNewClass .....	37

7.3.6.6	Methode: AcquireNewGraphicalDocumentFeatures...	37
7.3.6.7	Methode: AcquireNewData .....	38
7.3.7	DAU Spezialisten .....	38
7.3.7.1	Aufgaben zur Dokumenterfassung .....	39
7.3.7.2	Aufgaben zur Informationsextraktion.....	40
7.4	Kritikalitäten / Funktionen-Matrix.....	42



# 1 Allgemeines

## 1.1 Übersicht

Das vorliegende Dokument für das Projekt *Virtual Office* [VirtOff 97] stellt die Anforderungen an das zu entwickelnde System zusammen und beurteilt sie bei Bedarf.

## 1.2 Abkürzungen

DAU: Document Analysis and Understanding

WFMS: Workflowmanagementsystem

## 1.3 Definitionen

- System: Im folgenden Dokument versteht man hierunter das zu entwickelnde Gesamtsystem, das aus zwei Subsystemen besteht. Hierbei handelt es sich zum einen um ein kommerzielles WFMS, das erweitert und spezifisch adaptiert werden muß, daneben wird ein zu entwickelndes DAU-System beschrieben.
- SWPÄ: Software-Produktion und -Änderung, d. h. sämtliche Prozesse von Erstellung und Änderung der Software
- WFMS: Ein kommerzielles Workflowmanagementsystem (optional inklusive einer Geschäftsprozeß/Workflowmodellierungskomponente), das hinsichtlich der in diesem Dokument spezifizierten Anforderungen erweitert wird.
- Geschäftsprozeß: Der Geschäftsprozeß ist als höchste organisatorische Ebene für die Dauer einer Geschäftsabwicklung mit einem Kunden oder Lieferanten zu verstehen. Grundsätzlich läßt sich zwischen wertschöpfenden und nicht wertschöpfenden Prozessen unterscheiden. Erstere sind entscheidend für den Geschäftserfolg, da sie mit Umsätzen und Erlösen unmittelbar zusammenhängen. Nicht wertschöpfende Geschäftsprozesse sind lediglich indirekt für den Geschäftserfolg wirksam, da sie lediglich der internen Verwaltung dienen. Geschäftsprozesse enthalten sowohl manuelle als auch automatisierbare/rechnergestützte Prozeßanteile.
- Workflow[prozeß][definition]: Der Teil des Geschäftsprozesses, der aus einem Netzwerk automatisierbarer Tätigkeitsfolgen (Workflowaktivitäten) besteht.

- Workflow[prozeß]instanz: Die zur Laufzeit von einem WFMS verwaltete Instanz eines Workflows, die sich aus dem zugehörigen generischen Workflow durch die konkrete Variablenbelegung ergibt.
- Workflowaktivität: Ein logischer Schritt eines Workflows, der aus einer oder mehreren Workflowaktionen bestehen kann.
- DAU-Aktivität: Ein logischer Schritt eines Workflows, der aus einer oder mehreren DAU-Aktionen bestehen kann.
- Workflowaktion: Eine elementare, nicht teilbare Aktion, die durch Kompetenz/Rollenvergaben in der Aufbauorganisation potentiellen Akteuren zugeordnet ist.
- DAU-Aktion: Eine elementare, nicht teilbare Aufgabe, die durch Kompetenz/Rollenvergaben in der Aufbauorganisation (oder im Strategiewissen der DAU-Konfigurationseinheit) potentiellen DAU-Spezialisten zugeordnet ist, da sich für diese Aufgabe der Einsatz von DAU-Verfahren anbietet.
- Dokumentverifikation: Eine Workflowaktion, bei der über die übliche Rollenzuordnung die notwendige Kompetenz (z. B. Posteingangserfassung, Formularfeldverifikation) für diese Aufgabe sichergestellt wird. Die Aktion dient der Überprüfung und notfalls Änderung der Ergebnisse von DAU-Aktionen.
- Online-Teaching: Eine Workflowaktion, die im Bedarfsfall (Versagen der automatischen Dokumentanalyse) ausschließlich von einem geschulten Sachbearbeiterkreis zum Zwecke der Akquisition/Verfeinerung von Dokumentanalysewissen durchgeführt wird.
- Spezialist: Im vorliegenden Dokument werden unter dem Begriff Spezialist immer Dokumentanalysemodule (z. B. Segmentierung, Parser, ...) verstanden. Dabei ist es vom Kontext abhängig, ob es sich um einen generischen Spezialisten mit mehreren Wissensbasen für verschiedene Aufgaben handelt oder ob es sich um einen instantiierten Spezialisten mit genau einer Wissensbasis für die Lösung einer spezifischen Aufgabe handelt.
- Firmenwissen (corporate knowledge) bezeichnet Wissen über die Geschäftsprozesse und Datenstämme einer Firma (z. B. Aufbauorganisation, Rollen, Kunden, Produkte).

- Dokumentwissen (document knowledge) bezeichnet sowohl strukturelles als auch relationales Wissen über die Dokumente der Domäne (z. B. Layout-Objekte, Logik-Objekte, Inhalte).
- Spezialistenwissen (specialist knowledge) bezeichnet das prozedurale Wissen im Algorithmus des Spezialisten als auch das deklarative Wissen des Spezialisten (z. B. Grammatikregeln, Parameter). Wenn der deklarative Wissensteil eher eine allgemeine Form hat und eine Nutzung auch durch andere Spezialisten sinnvoll scheint, findet er sich im Dokumentwissen wieder.
- Strategisches Wissen (strategic knowledge) beschreibt den Analyseablauf (z. B. auf welchen Daten und in welchem Kontext geschieht der nächste Analyseschritt).
- Laufzeitwissen (runtime knowledge) enthält Zwischenergebnisse der Dokumentanalyse und offene Workflowinstanzen.
- Off-Line-Learning: Hierunter wird eine Form des Lernens verstanden, die in der Konfigurationsphase stattfindet, auf größeren Dokumentmengen arbeitet und für das Füllen der Wissensbasen mancher Spezialisten unbedingt notwendig ist.

Die verwendete Workflow-Terminologie orientiert sich an [WfMC 97].

## 1.4 Referenzierte Dokumente

[VirtOff 97] Proposal *Virtual Office*, DFKI internes Dokument, 1997.

[Lutzy 97] Ottmar Lutzy, C++-Entwicklungsrichtlinien, DFKI internes Dokument, 1997.

[WfMC 97] Glossary WorkflowManagementCoalition:

<http://www.aiai.ed.ac.uk/project/wfmc/>.

[Baumann et al. 97] Stephan Baumann, Michael Malburg, Harald Meyer auf'm Hofe, Claudia Wenzel. From Paper to a Corporate Memory - A First Step. Workshop on Knowledge-Based Systems for Knowledge Management in Enterprises, KI-97, Freiburg, Germany, September 1997, DFKI Document D-97-03.



## **2 Zweck des Dokuments**

Dieses Dokument beschreibt die Anforderungen an das zu entwickelnde System auf oberstem Abstraktionsniveau. Dabei ist dem Forschungscharakter des Gesamtsystems insoweit Rechnung getragen, als keine absoluten Anforderungen an spezifische Qualitätsmerkmale gestellt sind.



## **3 Analyseergebnisse**

### **3.1 Ist-Aufnahme/-Analyse**

#### **3.1.1 Ist-Aufnahme**

Workflowmanagementsysteme werden im Bürobereich zur effizienten Geschäftsprozessabwicklung verstärkt eingesetzt. Hierbei ist die Handhabung von papierintensiven Vorgängen in hohem Maße abhängig von einer Identifikation und Aufbereitung der in den Dokumenten enthaltenen Informationen. Ziel des Projektes *Virtual Office* ist daher die Erforschung und Entwicklung innovativer Lösungen für die automatische Dokumentanalyse und deren Integration in Workflow-Managementsysteme. Aufgrund des zu erwartenden Innovationspotentials hat der Auftraggeber (BMBF) ein großes Interesse an der Entwicklung eines prototypischen workflowzentrierten Dokumentanalyse-Systems.

#### **3.1.2 Ist-Analyse**

Obwohl WFMS die effiziente Behandlung von Geschäftsprozessen erlauben, fehlen Konzepte für eine transparente Informationseinbettung aus informellen Dokumenten. Daher besteht hier ein hoher Bedarf, um eine einheitliche, elektronische Verarbeitung geschäftsprozessrelevanter Informationen zu gewährleisten. Die Nutzbarmachung von *Corporate Knowledge* soll hierbei zur Steigerung der Erkennungsqualität von Dokumentenanalyseverfahren führen. Durch Einschränkung des aktuellen Kontextes – etwa in Form aktuell offener Vorgänge – kann außerdem eine Erhöhung der Erkennungs-Präzision erreicht werden. Domänenunabhängigkeit soll durch den Entwurf und die Implementierung modularer und konfigurierbarer Dokumentanalyse-Spezialisten erreicht werden. Da Off-Line-Lernen häufig zeitintensiv und massendatenabhängig ist, sollen On-Line-Lernverfahren eine Ergänzung dazu bilden und den unmittelbaren und adaptiven Einsatz solcher Systeme ermöglichen. Im Bereich der Bildverarbeitung sollen Grauwert- und Farbbilder auf geeignete Merkmale zur Verbesserung der Texterkennung und -analyse untersucht werden.

### **3.2 Bedrohungs- und Risikoanalyse**

Entfällt.



## 4 Allgemeine Anforderungen an das System

### 4.1 Gesamtfunktion des Systems

Das zu entwickelnde Gesamtsystem soll aus frei strukturierten Dokumenten Informationen extrahieren und workflow-kompatibel aufbereiten.

Hierzu wird *Corporate Knowledge* eingesetzt, welches in Workflows und Datenbankrelationen repräsentiert sein wird. Enthaltene Dokumentwissen muß geeignet aufbereitet und dann von den Spezialisten genutzt werden.

Das System soll domänenunabhängig einsetzbar sein und daher aus Wirtschaftlichkeitsgründen modular und konfigurierbar aufgebaut sein. Die Domäne wird im Bereich papierintensiver, administrativer Vorgänge angesiedelt sein.

Die unterschiedlichen Spezialisten/Systemkomponenten sollen on-line-lernfähig sein, im Bedarfsfall können zusätzlich Off-line-Lernverfahren eingesetzt werden.

Farb- und Grauwertbilder sollen als Dokumenteingabe so verarbeitet werden, daß bei Texterkennung und -analyse bessere Erkennungsraten als bei Binärbildern erreicht werden.

Faxe sollen neben (hochauflösend) eingescannten Dokumenten ebenfalls verarbeitet werden können.

### 4.2 Anforderungen hinsichtlich der Benutzung/Nutzung des Systems

Verschiedene Anwenderklassen werden mit der Benutzung des Systems auf den unterschiedlichsten Ebenen konfrontiert. Hierbei kann man typische Rollen unterscheiden, aber auch eine grobe Zuordnung des zu erwartenden Personenkreises, der aufgrund seiner Kompetenz für solche Rollen in Frage kommt, umreißen.

1. Allgemeines Workflow-Design: Diese Rolle wird typischerweise von einer mit dem Geschäftsprozeß vertrauten Person übernommen. Sie entwirft den Workflow und sollte dabei von einem graphischen Tool unterstützt werden.

2. Workflow-Anreicherung: Um einen vorhandenen Workflow gemäß unseren Vorstellungen um Dokumentanalysewissen anzureichern, soll eine einfache Schnittstelle realisiert werden. Hiermit kann ein geschulter Personenkreis entsprechende dokumentspezifische Anreicherungen durchführen.

3. Ausführen von Workflow-Aktionen: Durch Interaktion mit dem System werden die erforderlichen Arbeitsschritte von den Sachbearbeitern durchgeführt.

4. Dokumentverifikation: Die Dokumentverifikation als Workflow-Aktion kann vom gleichen Sachbearbeiterkreis wie in 3. ausgeführt werden, jedoch nicht zwingenderweise. Diese Aktionen können auch in einer vorgelagerten, zentralen Bearbeitung des Posteingangskorbs von einem zugeordneten Erfasser durchgeführt werden.

5. On-Line-Teaching: Der Benutzer benötigt eine einfach zu bedienende, einheitliche Schnittstelle zum On-Line-Teaching aller Spezialisten. Es ist davon auszugehen, daß diese Aktivitäten von einem autorisierten Personenkreis auf Sachbearbeiterebene durchgeführt werden, da entsprechendes Hintergrundwissen notwendig ist.

6. Systemkonfiguration: Die Systemkonfiguration beinhaltet die Erstellung eines lauffähigen Systems auf Basis des Workflows und der Spezialisten. Der Systemoperator erstellt oder übernimmt für jede Aktivität manuell eine Default-Konfiguration oder erwählt die automatische Konfiguration und kann diese im Bedarfsfall ändern, wobei er dann über Monitoringtools für die Dokumentanalyse verfügt. Solche Änderungen dürfen zu jedem Zeitpunkt nur von einem Systemoperator gemacht werden. Diese Person muß daher mit den Fragen der Dokumentanalyse vertraut sein. Weiterhin ist diese Person für die Aktualisierung der Laufzeitspezialisten um das „geteachte“ neue Wissen verantwortlich. Das Aktualisieren der betroffenen Spezialisten erfolgt transparent für den Benutzer zu einem definierten Zeitpunkt, nicht während der Laufzeit der betroffenen Spezialisteninstanz.

7. Off-line Teaching: Das Offline Teaching sollte einfach organisiert sein und zentral vom Systemoperator durchgeführt werden.

### **4.3 Anforderungen hinsichtlich der Realisierung interner**

## **Schnittstellen**

Die im Forschungsbereich definierten C++-Richtlinien [Lutzy 97] sind bei der Realisierung einzuhalten.

Eine Portierbarkeit auf Solaris und WindowsNT-Plattformen sollte gewährleistet werden. Durch Verwendung der QT und STL-Libraries sollte dies mit geringem Aufwand möglich sein.

### **4.4 Kritikalität des Systems**

Entfällt.

### **4.5 DV-Sicherheit**

Entfällt, da keine Risikoanalyse und Bedrohungsanalyse durchgeführt wurde.

### **4.6 Anforderungen an sonstige Qualitätsmerkmale des Systems**

Es soll geprüft werden, ob sich die Verwendung von *Corporate Knowledge* hinsichtlich der Erkennungsqualität von Dokumentenanalyseverfahren qualitätssteigernd auswirkt. Um einen solchen Vergleich durchführen zu können, soll die Dokumentanalyse auch mit einem initialen Workflow lauffähig sein, der eigentlich keinerlei Informationen enthält.

Die Granularität der im Workflow spezifizierten Aktionen, wozu auch die Schritte der Dokumentanalyse gehören, ist variabel, wodurch eine dynamische Verlagerung von Steuerungswissen möglich ist.

Das Dokumentwissen soll zeiteffizient verwendet werden. Dazu ist insbesondere sorgfältig zwischen den beiden Möglichkeiten der Wissenskompilation zur Generierungszeit bzw. der Interpretation zur Laufzeit abzuwägen.

Das System (hier die Konfigurations/Steuerungskomponente) soll nach Bedarf ressourcen- oder genauigkeits-optimierend arbeiten können. Ressourcen von besonderem Interesse sind hierbei Speicherplatz und Rechenzeit. Das Systemverhalten soll bezüglich einer Präferenz für Recall oder Precision vom Operator variabel spezifiziert werden können.

Um On-Line-Learning und mögliches Pipelining von Dokumenten in Batchjobs, d. h. eine parallele Verarbeitung, konsistent durchführen zu können, muß die Integrität der Parameter auf Spezialistenebene sichergestellt sein. Daher müssen während einer Dokumentanalyseaktivität die Parametersätze, die aus den Workflowinstanzen beschafft werden, für beteiligte Spezialisten konsistent sein.

## **5 Anforderungen an die Entwicklungs- und SWPÄ-Umgebung**

Die Entwicklung des Systems wird unter SUN/Solaris und/oder WindowsNT stattfinden. Als Programmiersprache wird C++ verwendet werden. Als Entwicklungsumgebung wird SUN Workshop Pro eingesetzt, bzw. in der WindowsNT-Welt Visual C++ 5.0.

Die Generierung, Pflege und Änderung der Workflowspezifikationen wird in Abhängigkeit vom eingesetzten WFMS, bzw. Workflowmodellierungstool erfolgen (SUN-Solaris, Windows NT). Zur Zeit wird der CSE-Prozeßdesigner hierzu verwendet, alternativ kann auch mit ADONIS gearbeitet werden.

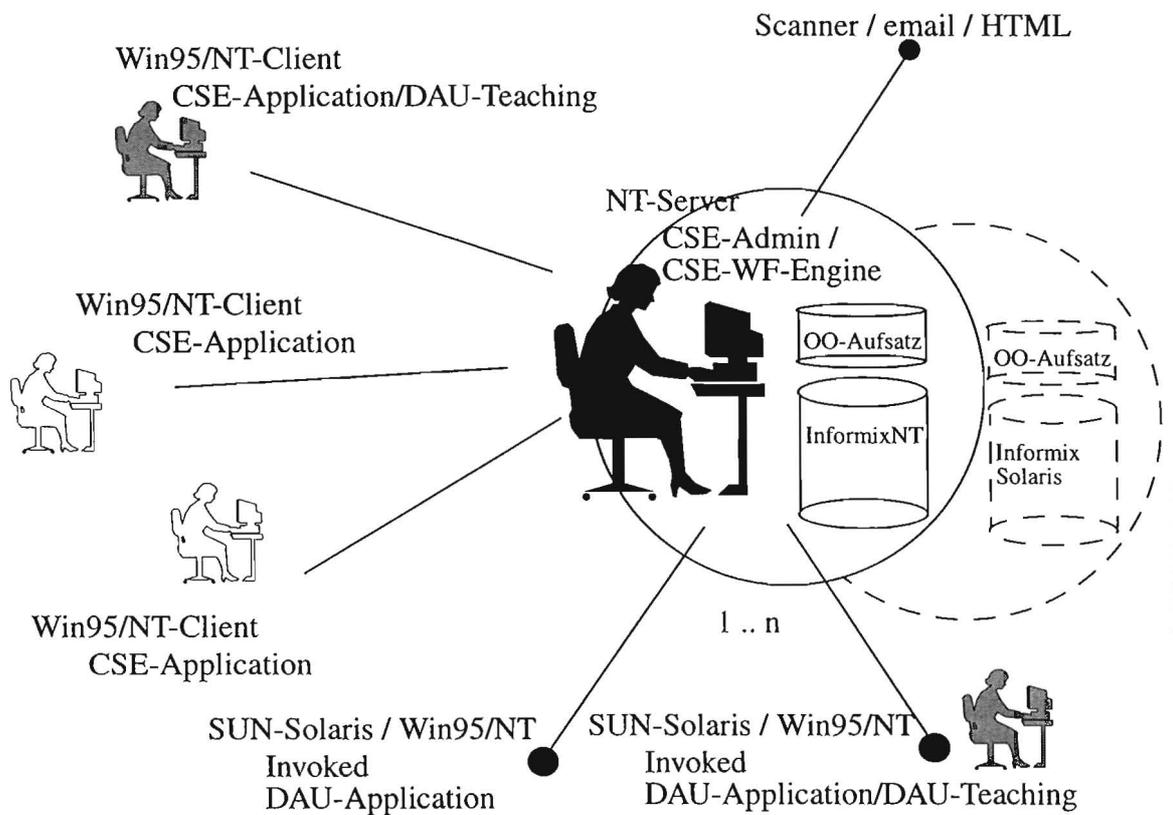


# 6 Anforderungen an die systemexternen Schnittstellen

## 6.1 Einsatzumgebung

### 6.1.1 Technische Einsatzumgebung

Das zu entwickelnde Gesamtsystem wird in eine technische Einsatzumgebung eingebettet, die für den Betrieb erforderlich ist. Zu erwarten ist hierbei ein heterogenes Netzwerk basierend auf SUN-Workstations unter Solaris und WindowsNT-Rechnern.



**ABBILDUNG 1: Einsatzumgebung von *Virtual Office***

Die Dokumenterfassung erfolgt mittels Faxmodem bzw. Farbscanner.

### **6.1.2 Organisatorische Einbettung**

Die unter 3.2. aufgeführten Rollen müssen gemäß der vorhandenen Organisationsstruktur und Personalressourcen belegt werden, um eine Einbettung des Gesamtsystems in die Organisation zu gewährleisten.

### **6.1.3 Nutzung**

Das System wird je nach Anwendungsdomäne vorkonfiguriert und dann vom Benutzer zur Vorgangsbearbeitung mit integrierter Dokumentanalyse eingesetzt.

## **6.2 Sonstige systemexterne Schnittstellen**

### **6.2.1 Übersicht**

Schnittstelle Workflow-Beschreibung/-Design

Schnittstelle Workflow-Anreicherung

Schnittstelle Workflow-Aktionen

Schnittstelle Dokumentverifikation

Schnittstelle On-Line-Teaching

Schnittstelle Off-Line-Learning

Schnittstelle Systemkonfiguration

Schnittstelle Testumgebung

Schnittstelle Dokumente

Schnittstelle Monitoring und Tuning

## 6.2.2 Einzelbeschreibungen

### 6.2.2.1 Schnittstelle *Workflow-Beschreibung/-Design*

Hierunter fällt die Beschreibung aller Aspekte des administrativen Workflows. Dazu gehören anfallende Aktionen, Aktoren, Rollen, Trigger, Dokumente, Gruppen, Menschen und Systeme.

### 6.2.2.2 Schnittstelle *Workflow-Anreicherung*

Hierunter fällt die Beschreibung aller Modellierungsaspekte bzgl. des Workflows, die für die Dokumentanalyse relevant sind. Dazu gehört die Beschreibung von typischen Dokumenten der Domäne, die aus einer generischen Dokumentbeschreibung mit ROIs und entsprechenden Attributen für Geometrie-/Layoutinformationen, Farbbinarisierung, spezialisierte OCR, Wörterbücher, assoziative DB-Abgleiche u. s. w. besteht. Dies beinhaltet auch Informationen, die durch On-Line-Teaching bereitgestellt wurden (z. B. Aufbau der Auftragsnummer der Uni) und von einzelnen DA-Spezialisten benötigt werden. Außerdem wird der Workflow um Informationen zur Aktivierung und Eignung von Dokumentanalysespezialisten angereichert. Dies geschieht in Form von allgemeinverständlichen Kapselbeschreibungen.

### 6.2.2.3 Schnittstelle *Workflow-Aktionen*

Primär erfolgt hier eine Benachrichtigung der Sachbearbeiter über anliegende Aktionen und eine umgekehrte Benachrichtigung des Systems bei Erledigung. An dieser Stelle wird weitestgehend die Funktionalität des WFMS genutzt. Zusätzlich erfolgt ein Aufruf des Subsystems, um die DAU-Aktivitäten ausführen.

### 6.2.2.4 Schnittstelle *Dokumentverifikation*

Über eine graphische Benutzerschnittstelle kann der Sachbearbeiter folgende Teile der Analyseergebnisse des DAU-Systems ändern. Hierbei ist für die Verifikation der freie einfache Zugriff auf das Brutto-Image wünschenswert:

- Korrektur der Texterkennungsergebnisse (mandatory)
- Korrektur von Segmentierungsergebnissen bei der Formulareerkennung (nice-to-have)

- Korrektur der logischen Zuordnung (nice-to-have)
- Korrektur der extrahierten Information via Datenbanken bzw. „intelligente Menüs“, z. B. Adressen via Kundenstamm-DB (mandatory)
- Korrektur der Vorgangszuordnung (mandatory)

#### 6.2.2.5 Schnittstelle *On-Line-Teaching*

Der autorisierte Benutzer kann über eine graphische Schnittstelle die folgenden Daten eingeben:

- Angabe und Benennung (Logiklabel, Patternnamen, ...) der Region, in der ein Muster zu finden ist, sowie die Markierung der zu extrahierenden Teile und die Markierung fester Bestandteile, nach denen gesucht werden kann
- Synonyme und Hyper-/Hyponyme
- Angaben über mögliche Farben des Vorder-/Hintergrunds und die Verteilung von Farben im Dokument
- Leerformularerstellung
- Referenztabellenerstellung
- Zuweisung von Dokumentklassen (z. B. Nachrichtentyp, Produktkategorie, Formular-typ/-nummer) zum gesamten Dokument
- Eingabe neuer Wissensportionen wie Adressen, Leute, Firmen, Produkte.

#### 6.2.2.6 Schnittstelle *Off-Line-Learning*

Hier ist darauf zu achten, daß eine genügend große Dokumentmenge zum Lernen bereitsteht und dem System bekannt gemacht wird. Name und Lokalisierung dieser Mengen sind für den Spezialisten bzw. die Konfigurationsphase zugänglich. Falls dies nicht möglich ist, liefert der betroffene Spezialist kein Ergebnis.

#### 6.2.2.7 Schnittstelle *Systemkonfiguration*

- Integration und Entfernen dokumentanalyse-spezifischer Software-Module

- Spezifikation der verfügbaren Systemressourcen z. B. Rechenkapazität, Speicherkapazitäten klassifiziert nach Eigenschaften, die für die Software-Module der Dokumentanalyse relevant sind.
- Vereinbarung der im Workflow möglichen DAU-Aktivitäten. Dazu gehören
  - die Beziehung zwischen der DAU-Aktivität und ihrer Realisierung als Software-Modul,
  - Ein- und Ausschalten des Dokumentpipelining
  - Wahl der Recall/Precision-Präferenz,
  - Erstellen einer Standardkonfiguration für eine DAU-Aktivität,
  - spezifische Eigenschaften wie Parametersätze der an der Realisierung beteiligten DAU-Spezialisten und Zugriff auf über den Workflow zu beziehende Daten.

#### 6.2.2.8 Schnittstelle *Testumgebung*

Zwei verschiedene Testumgebungen sollen bereitgestellt werden:

- Testumgebung für Einzelkomponenten: Hierbei ist zu beachten, daß Einzelkomponenten vor Bereitstellung des Gesamtsystems getestet werden können, d. h. alle Eingaben an Schnittstellen der Komponente sollten entweder vorhanden sein oder simuliert werden können. Hier besteht als Überbrückung die Möglichkeit, auf OMEGA Datenstrukturen zuzugreifen, bis die fehlenden Eingaben/Komponenten vorhanden sind.
- Testumgebung für das Gesamtsystem: Das Gesamtsystem wird in einem heterogenen Solaris/WindowsNT Netz getestet. Hierzu werden Testvorgänge incl. aller enthaltenen Dokumente erstellt und in Kombination mit Rücksetzen des Systems mögliche Workflow-Pfade unter verschiedenen Umgebungssituationen durchgespielt. Hierzu wird die Funktionalität wie im Echtbetrieb benötigt und parallel ein gezieltes DAU-Monitoring/Tuning durchgeführt. Hierzu sollen auch Backup- und Recoveryfunktionen geschaffen werden.

Geeignete Bewertungsmaße, sowohl für die Einzelkomponenten als auch für das Gesamtsystem, sollten angewendet werden, wo dies möglich ist.

Zum Testen sind Testdokumente notwendig. Diese sollen als „gute“ Repräsentanten aller möglichen Beispiellklassen ausgewählt werden. Zur Bewertung der System-/Komponentenleistung müssen zu den Testfällen GroundTruth Mengen bereitgestellt werden. Diese umfassen ASCII Text, Layout- bzw. Formularzonen, logische Label, Inhalte etc.

#### 6.2.2.9 Schnittstelle *Dokumente*

Die Dokumentschnittstelle umfaßt sowohl eingehende Dokumente als auch die Ablage/ Archivierung dieser Dokumente. Dokumente werden über zwei Arten von Medien erfaßt:

- Faxe über Faxmodem
- Briefe werden mit 300 dpi eingescannt. Die Bildpunkte können binär, grauwertig und farbig sein.

Die Dokumente können mehrseitig sein. Originalbild und bereinigtes Bild sollen beide der Analyse zur Verfügung stehen.

#### 6.2.2.10 Schnittstelle *Monitoring und Tuning*

Monitoring soll das Entdecken von Schwachstellen, bei Programmabstürzen das Lokalisieren der Fehlerquellen und eine bessere Verständlichkeit des Prozesses ermöglichen. Dabei soll lediglich die Überwachung ermöglicht werden, Eingriffe in das Systemverhalten jedoch nicht erlaubt sein. Dazu ist eine komplette Übersicht über den Daten- und Kontrollfluß im DAU bzw. im WFMS erforderlich. Statusinformation soll jederzeit abfragbar und analysierbar sein. Das Format soll benutzerfreundlich sein. Mögliche Anfragen sind z. B. das Monitoring aller Events, aller Vorgänge, eines bestimmten Vorganges, einer bestimmten Softwarekomponente. Mögliche Zustände sind z. B. running, waiting, blocked

Beim System-Tuning sollen Daten- und Kontrollfluß vom Systemoperator verändert werden dürfen. Womöglich ergibt sich daraus die Möglichkeit, eine Art On-Line-Learning auch für die Kontrolle des Systems durchzuführen.

## **7 Funktions- und datenorientierte fachliche Anforderungen an das System aus objektorientierter Sicht**

### **7.1 Anwendungsszenarien**

#### **7.1.1 Off-Line Lernphase für einzelne Spezialisten**

Bestimmte DAU-Spezialisten müssen auf einer Trainingsmenge lernen, bevor ihre Wissensbasen zum effektiven Einsatz genutzt werden können (z. B. Trainieren von Gewichten, Lernen von Regeln, etc.). Diese Trainingsphase wird in die Konfigurationsphase integriert, so daß der Administrator den Trainingsvorgang initiiert und kontrolliert. Dazu muß er zunächst die benötigten Trainingsdaten zur Verfügung stellen und kann dann gezielt einzelne Spezialisten „lernen lassen“. Wichtig ist, daß die Art der für den Spezialisten benötigten Trainingsdaten für den Administrator verständlich und zugreifbar dokumentiert ist.

#### **7.1.2 Konfigurationsphase**

Der Systemoperator hat eine Liste der im Workflow-Modell vorkommenden Aktionen, die sich mit Aufgaben der Dokumentanalyse befassen, im Betrieb also zu einem Aufruf der Kontrollkomponente führen sollen. Die Konfigurierungskomponente bietet dem Systemoperator nun einen Dialog an, in dem er beginnend vom allgemeinsten Begriff des strategischen Wissens, der DAU-Aktivität, Spezialisierungen, Dekompositionen, Parametrisierungen aber auch konkrete Aggregierungen spezifizieren kann, um die auszuführende DAU-Aktivität zu beschreiben.

Hält der Operator die DAU-Aktivität für ausreichend beschrieben, so initiiert er die Berechnung einer technischen Realisierung dieser Komponente. Dies hat zwei Gründe: Zum einen muß hinreichend nachgewiesen werden, daß eine technische Realisierung der gewünschten DAU-Aktivität besteht. Zum anderen neigen Menschen bei der Eingabe von Spezifikationen häufig zur Unterspezifikation des Problems. Derartige Unterspezifikationen können durch Vorlage von Lösungen bemerkt werden. Erscheint diese Lösung nämlich als unzureichend, so müssen zusätzliche Angaben zur Problemspezifikation gemacht werden, die zumindest die Kritikpunkte an die vorgelegte Lösung umfassen.

Nach der Konfigurationsphase besteht für jede die Dokumentanalyse betreffende Workflow-Aktivität ein Eintrag, der eine technische Realisierung der geforderten DAU-Aktivität beschreibt. Zum anderen besteht eine abstrakte Beschreibung der DAU-Aktivität in Termini des strategischen Wissens. Diese kann ggf. verwendet werden, um während des Betriebs andere technische Realisierungen der Aktivität zu berechnen.

### **7.1.3 Analysephase**

Zur Laufzeit des WFMS findet die automatische Analyse neu eingescannter Dokumente transparent für den Benutzer statt. Die vorhergehende Konfigurationsphase stellt sicher, daß zu allen DAU-Aktivitäten technische Realisierungen bereitstehen. Die Zeitpunkte der Analyse wurden vorab in der Workflow-Modellierung spezifiziert, wobei eine zentrale Verarbeitung des gesamten Eingangspostkorbs vorstellbar ist, aber auch dezentrale, gezielte DAU-Aktivitäten modelliert werden können.

Die zur Analyse benötigten Spezialisten werden zur Laufzeit von einer zentralen Steuerungskomponente mit den Kontexten der offenen Workflow-Instanzen versorgt und aktiviert. Da diese Komponente über das strategische Wissen verfügt, gewährleistet sie auch eine effiziente Analyse der vorliegenden Dokumente – bei einer Vielzahl von offenen Vorgängen und eingehenden Dokumenten ist eine möglichst redundanzfreie Verarbeitung erstrebenswert. Kann eine Erschließung der gewünschten Dokumentinhalte nur partiell oder gar nicht erfolgen, müssen entsprechende Maßnahmen zum On-Line-Teaching angestoßen werden.

Im Stand-Alone-Betrieb wird die Analyse von Dokumenten auch ohne die WFMS-Einbettung ermöglicht. Die Modellierung enthält hierbei entweder alle gewünschten DAU-Aktionen oder eine DAU-Aktivität für die Gesamtanalyse. Die Steuerungskomponente aktiviert wiederum die Spezialisten, allerdings ohne Kontextunterstützung.

### **7.1.4 Verifikation analysierter Dokumente**

Die Verifikation der analysierten Dokumente darf vom „einfachen“ Benutzer durchgeführt werden. Wie in Kapitel 6.2.2.4 bereits beschrieben, umfaßt sie mindestens

- die Korrektur der Texterkennungsergebnisse,



Beispiel für die Aktualisierung offener Vorgänge ist die korrekte Zuordnung zum richtigen Vorgang durch Korrektur des Datums bzw. der Auftragsnummer.

Der Zeitpunkt einer möglichen Verifikation wird in der Workflowspezifikation als Aktion vorab definiert.

### **7.1.5 Aktualisieren von Workflow- und Dokumentdaten**

Bei der Verifikation eingehender und bereits partiell analysierter Dokumente durch einen Sachbearbeiter/Benutzer (vgl. Kapitel 6.2.2.4) kann es vorkommen, daß als Ursache des Nichterkennens bestimmter Dokumentinhalte durch das DAU-System Lücken oder Fehler in den Daten- oder Wissensbasen ausgemacht werden. Für diesen Fall soll gewissen „autorisierten Benutzern“ die Möglichkeit geboten werden, über eine weitere Schnittstelle diese Mängel zu beheben (vgl. Kapitel 6.2.2.5).

Hierbei ist zu beachten, daß diesem anwendungsorientierten Benutzer keine besonderen Kenntnisse, z. B. intern verwendeter Datenstrukturen, abverlangt werden. Um dies zu erreichen, sollen etwaige Änderungen unter Anlehnung an die Analysedaten eingegeben werden. Dazu ein Beispiel: Rührt ein Mangel der Informationsextraktion daher, daß ein relevantes Pattern nicht gefunden wurde, so geht der Benutzer in folgenden Schritten vor. Zuerst markiert er auf dem Dokumentbild das besagte Pattern durch Anklicken der dazugehörigen Wörter. Danach teilt er dem System mit, wie dieses Pattern heißt und – gegebenenfalls – was seine besonderen Merkmale, etwa unveränderliche Wörter im Pattern, sind. Dieser letztgenannte Teil kann ebenso bereits Teil des darauffolgenden, systeminternen Lernvorgangs sein. Schließlich gibt der Benutzer noch die übergeordneten (Layout-) Regionen an, in welchen dieses Pattern typischerweise zu finden ist.

Auf die prinzipiell gleiche Weise sollen, wie bereits in Kapitel 6.2.2.5 ausgeführt, die semantischen Inhalte des Systemlexikons, insbesondere Synonyme und Hyper-/Hyponyme, gepflegt werden können. Ebenso sind Angaben über typische Farbverteilungen im Dokument möglich, es können Leerformulare erstellt, Dokumentklassen zugeordnet und neu entworfen sowie weitere Wissensportionen innerhalb der Firmendomäne erfaßt werden.

Bei einem kommerziell eingesetzten System wäre ein entsprechend hoher Aufwand bei der Oberflächenprogrammierung für diesen Anwendungsfall sicherlich angebracht. Im Rahmen des geplanten Forschungsvorhabens soll eine entsprechende Benutzerschnittstelle einerseits die genannten Funktionalitäten bieten, andererseits jedoch nicht unbedingt den derzeitigen Standards hinsichtlich Design und Komfortabilität genügen: ein pragmatisches Vorgehen, bei dem der Nachweis von Machbarkeit und Sinn der Aktion im Vordergrund stehen, ist zu präferieren. Der Zeitpunkt dieser Ergänzungen wird in der Workflowspezifikation als Aktion vorab definiert.

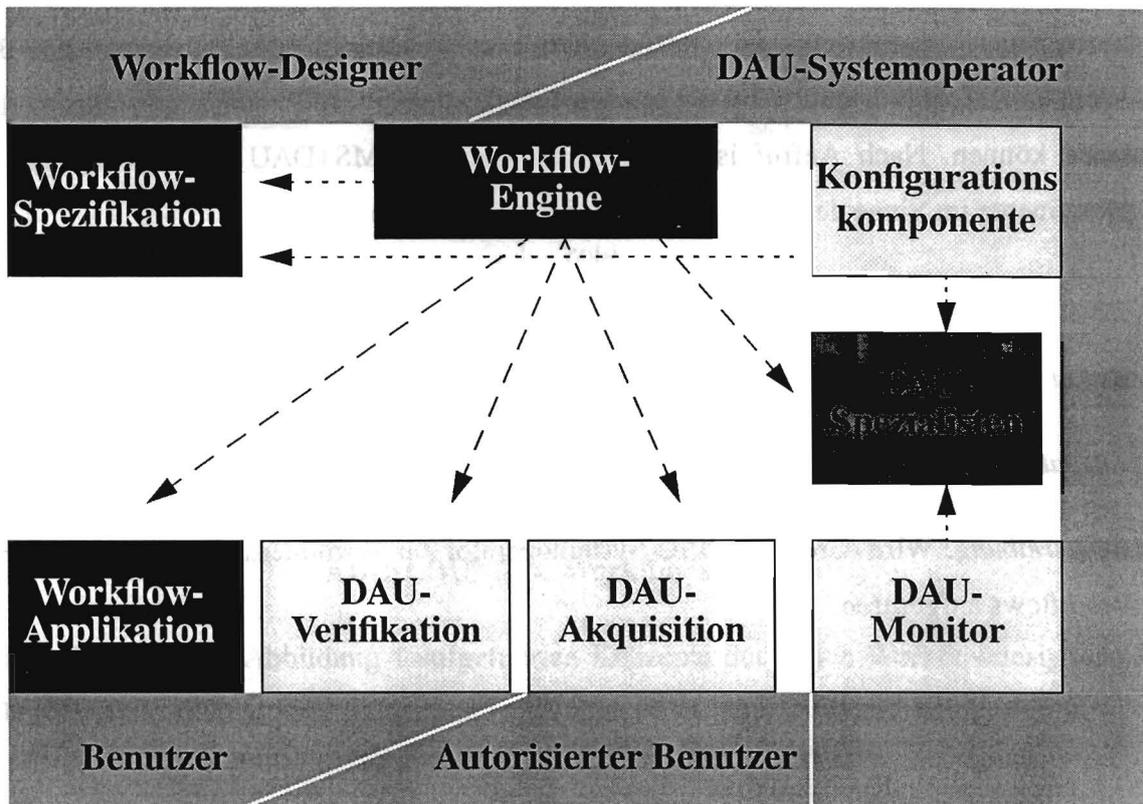
## 7.2 Klassenkategorien und Rollen des Problembereichs

Rollen:

- Workflow-Designer
- DAU-Systemoperator
- Autorisierter Benutzer
- Benutzer

Klassenkategorien:

- Workflow-Engine
- Workflow-Spezifikation
- Workflow-Applikation
- Konfigurationskomponente
- DAU-Monitor
- DAU-Verifikation & Akquisition
- DAU-Spezialisten



- ← - - - - Zugriff auf Daten
- ← - - - - Aktivierung
- implementierte Klassenkategorie
- Rolle
- zu entwickelnde Klassenkategorie
- zu entwickelnde Klassenkategorie ohne Rollenzugriff

ABBILDUNG 3: Klassenkategorien und Rollen

## 7.3 Klassenkategorien- und Rollenbeschreibungen

### 7.3.1 Workflow-Engine

#### 7.3.1.1 Methode: StartEngine

Argumente: WorkflowSpezifikation(en)

Ausgabe:

*Beschreibung:* Wird durch den DAU-Systemoperator nach abgeschlossener Konfigurationsphase aufgerufen. Neben der eigentlichen Spezifikation des Workflows müssen hierzu auch die notwendigen systemtechnischen Voraussetzungen (OrganisationsDB, WorkflowDB, etc.) erfüllt sein, die je nach eingesetztem WFMS produktspezifisch variieren können. Nach Aufruf ist das Gesamtsystem WFMS+DAU bereit, eingehende Dokumente im Sinne des oder der Workflows zu bearbeiten.

#### 7.3.1.2 Methode: StopEngine

*Argumente:* WorkflowSpezifikation(en)

*Ausgabe:*

*Beschreibung:* Wird durch den DAU-Systemoperator zur Terminierung eines (mehrerer) Workflows aufgerufen.

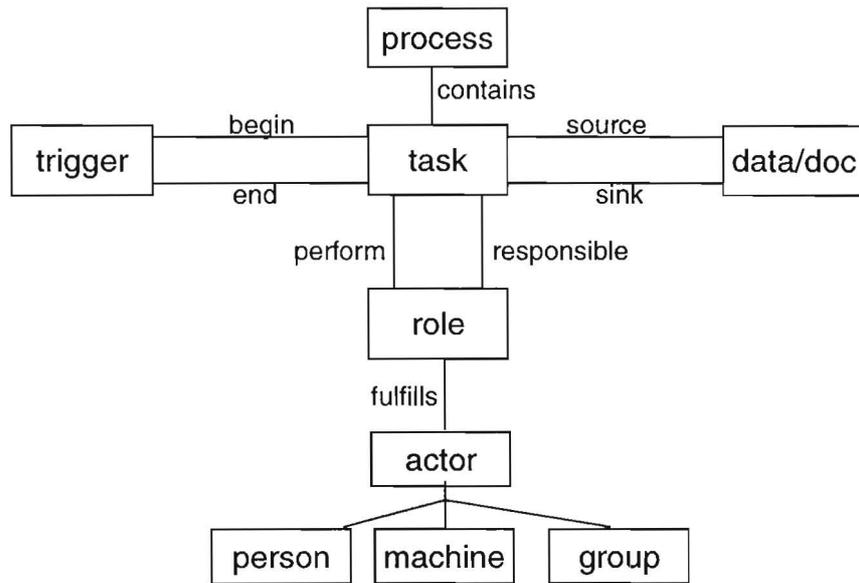
### 7.3.2 Workflow-Spezifikation

#### 7.3.2.1 Methode: NewWorkflow

*Argumente:*

*Ausgabe:* Ein vollständig spezifizierter Workflow im Sinne des zugrundeliegenden WFMS-Produkts.

*Beschreibung:* Wird durch einen Workflow-Designer aufgerufen. Hier wird in der Regel ein im Lieferumfang des WFMS enthaltenes graphisches Tool (hier: CSE-Prozeßdesigner, siehe Dokumentation zur Funktionalität) eingesetzt, das die entsprechenden Funktionalitäten zur Spezifikation der erforderlichen Prozeßstruktur zur Verfügung stellt.



**ABBILDUNG 4: Workflowkonzepte**

Dabei sind die in Abbildung 4 aufgeführten Konzepte durch den Workflowdesigner zu beschreiben (s. auch Kap. 3.2 in [Baumann et al. 97])

#### 7.3.2.2 Methode: ChangeWorkflow

*Argumente:* Ein zuvor im System spezifizierter Workflow.

*Ausgabe:* Ein geänderter Workflow

*Beschreibung:* Wird durch einen Workflowdesigner aufgerufen. Hier wird in der Regel ein im Lieferumfang des WFMS enthaltenes graphisches Tool (hier CSE-ProcessDesigner) eingesetzt, das die entsprechenden Funktionalitäten zur Änderung der erforderlichen Prozeßstruktur zur Verfügung stellt.

#### 7.3.2.3 Methode: EnrichWorkflow

*Argumente:* Ein zuvor im System spezifizierter Workflow.

*Ausgabe:* Workflow angereichert mit Dokumentwissen

*Beschreibung:* Wird durch einen Workflowdesigner, der sich hoffentlich mit Dokumenten auskennt, aufgerufen. Hier wird Dokumentwissen spezifiziert, was eine Anreicherung des Workflows darstellt. Es ist entweder eine zentrale Spezifikation von Workflow

und Dokumentwissen oder aber eine lokale Spezifikation von Dokumentwissen vorstellbar (siehe auch Kap. 3.2. in [Baumann et al. 97]).

*Bemerkung:* Hier ist die vorhandene Ausdrucksmächtigkeit für die Dokumenttypmodellierung in CSE-Workflow/Prozeßdesigner, bzw. WfMC-Referenzmodell/Interface1 noch genau zu evaluieren. Bekannt ist, daß Metadaten zu Dokumenten als auch zu Business-Cases (Sammlung von zusammengehörigen Dokumenten) dargestellt werden können: eindeutige ID zu Verwaltungszwecken, Intern/Extern, elektronisch/Papier, Absender und Empfänger (Adresse, Telefon, Fax, Kürzel), interner/externer Kommentar, Kundennummer, Grußformel, Posteingangsdatum, Absendedatum, selbstdefinierte Metadaten.

#### 7.3.2.4 Methode:DeleteWorkflow

*Argumente:* Ein zuvor im System spezifizierter Workflow.

*Ausgabe:*

*Beschreibung:* Wird durch einen Workflowdesigner aufgerufen. Hier wird in der Regel ein im Lieferumfang des WFMS enthaltenes graphisches Tool (hier CSE-Prozeßdesigner) eingesetzt, das die entsprechenden Funktionalitäten zum Löschen der angegebenen Workflowspezifikation enthält.

### 7.3.3 Workflow-Applikation

Die Klassenkategorie „Workflow-Applikation“ stellt die Funktionalität eines normalen Sachbearbeiter-Arbeitsplatzes bereit.

#### 7.3.3.1 Methode: GetOpenProcesses

*Argumente:* Benutzer

*Ausgabe:* Übersicht über die offenen Vorgänge des angegebenen Benutzers

*Beschreibung:* Wird durch einen Benutzer (Sachbearbeiter) aufgerufen. Der Benutzer sieht die Vorgänge in chronologischer Reihenfolge. Aus dieser Liste kann er sich einen Vorgang zum Bearbeiten auswählen.

### 7.3.3.2 Methode: WorkProcess

*Argumente:* Ein offener Vorgang

*Ausgabe:*

*Beschreibung:* Der Benutzer bearbeitet hier den Vorgang. Nach Erledigung wird der Vorgang gemäß der Vorgangsspezifikation zum nächsten Sachbearbeiter weitergeleitet.

### 7.3.3.3 GetStateOfProcess

*Argumente:* Ein Vorgang

*Ausgabe:* Status des Vorgangs

*Beschreibung:* Mit dieser Funktion hat der Benutzer (Sachbearbeiter) die Möglichkeit, den momentanen Zustand eines Vorgangs zu ermitteln. Er sieht die einzelnen Verarbeitungsstufen, sowohl die bereits erledigten als auch die noch zu erledigenden.

### 7.3.3.4 Methode: SetSubstitution

*Argumente:* Zwei Benutzer (Sachbearbeiter)

*Ausgabe:*

*Beschreibung:* Die Methode wird von einem Sachbearbeiter aufgerufen. Er bestimmt damit seinen Vertreter.

### 7.3.3.5 SendMail

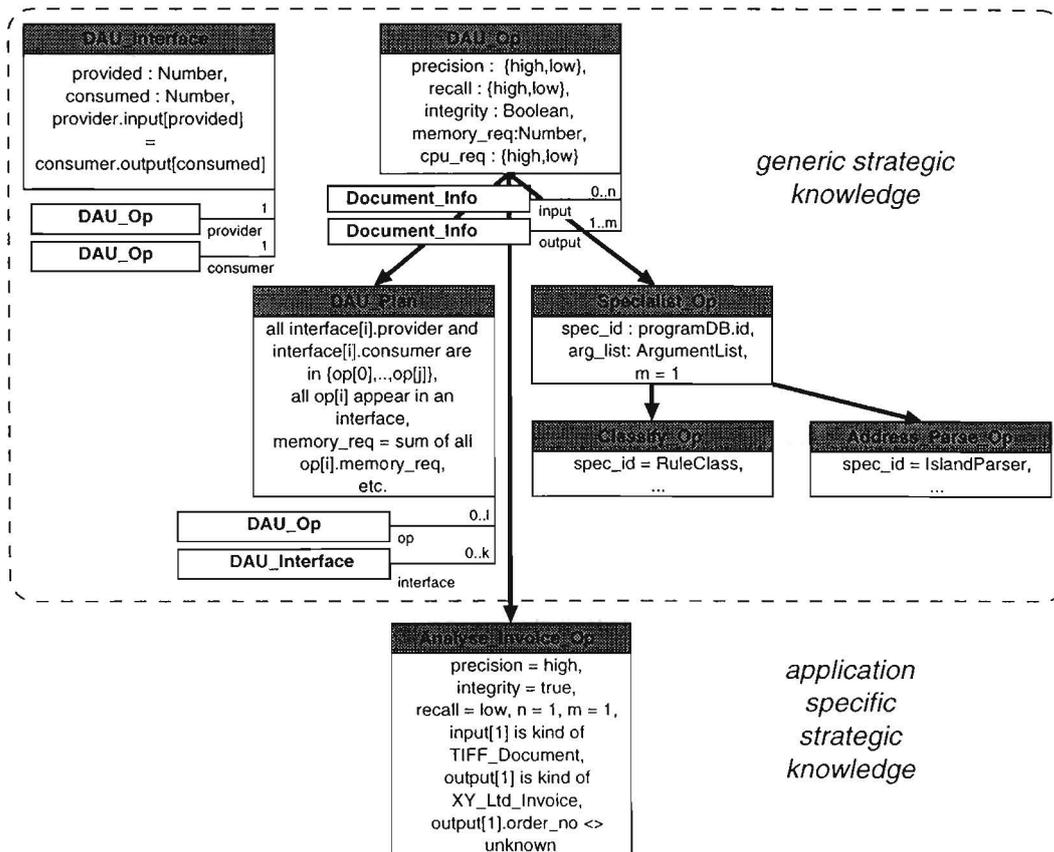
*Argument:*

*Ausgabe:*

*Beschreibung:* Diese Methode wird von einem Benutzer aufgerufen. Er kann mit dieser Methode beliebigen Personen zusätzliche Information zu einem Vorgang zukommen lassen.

### 7.3.4 Konfigurationskomponente

Abstrakte Anforderungen an die Konfiguration der DAU-Spezialisten werden mittels einer Konzepthierarchie spezifiziert, wie sie beispielhaft in Abbildung 5 angegeben ist. Daher besteht die Benutzerschnittstelle der Konfigurationskomponente im wesentlichen aus einem Konzepteditor, der zum einen die Beziehungen der verwendeten Konzepte



**ABBILDUNG 5: Bestandteile einer Konzepthierarchie zur Beschreibung von DAU-Aktionen**

anzeigen und zum anderen Elemente der verwendete Konzeptsprache einfügen kann. Bestandteile dieser Konzeptsprache sind explizite Subsumptionsbeziehungen, Attribute, Integritätsbedingungen (Constraints) zwischen Attributen und eine Partonomie. Ergebnis ist eine abstrakte Beschreibung spezieller DAU-Aktivitäten durch Mengen kompatibler Systemkonfigurationen. Diese enthalten die verwendeten DAU-Spezialisten inklusive ihrer „Verschaltung“ und Parametrisierung.

#### 7.3.4.1 Methode: ViewConcepts

*Argumente:* Datenbank der definierten Konzeptbeschreibungen

*Ausgabe:* Graphische bzw. Browser-Darstellung der taxonomischen Hierarchie der definierten DAU-Aktivitäten, ihrer Attribute und ihrer Komponenten

*Beschreibung:* Über diese Methode navigiert der Systemoperator in der Phase Systemkonfiguration (vgl. Kapitel 7.1.2), um sich einen Überblick über die bislang definierten DAU-Aktivitäten zu verschaffen und ggf. anwendungsspezifische Spezialisierungen durchführen zu können.

#### 7.3.4.2 Methode: AddConcept

*Argumente:* Eindeutiger Name eines neuen Konzepts einer DAU-Aktivität

*Ausgabe:* Veränderte Hierarchie der definierten Konzeptbeschreibungen

*Beschreibung:* In der Phase Systemkonfiguration (vgl. Kapitel 7.1.2) fügt der Systemoperator mittels dieser Methode ein neues Konzept (neue DAU-Aktivität) in die Datenbank ein.

#### 7.3.4.3 Methode: AddConstraint

*Argumente:* Konzept einer DAU-Aktivität, ein Vektor mit Attributnamen oder Verweis auf die Kardinalität einer Komponente, Beschreibung der Extension des Constraints

*Ausgabe:* Darstellung der Beschreibung des modifizierten Constraints

*Beschreibung:* Mittels dieser Methode führt der Systemoperator eine Integritätsbedingung zwischen Attributen des angegebenen Konzepts ein. Es können auch ererbte Attribute bezeichnet werden. Durch Verweis auf die Kardinalität einer Komponente (vgl. Kapitel 7.3.4.7) kann auch das Vorkommen einer Komponente in Abhängigkeit der Ausprägung anderer Attribute eingeschränkt werden. Wird auf ein Attribut einer Komponente verwiesen, so wird implizit für die Kardinalität  $k$  dieser Komponente das Constraint  $k > 0$  eingeführt. Die Integritätsbedingung wird in einer Constraintsprache angegeben.

#### 7.3.4.4 Methode: AddConfigurationManually

*Argumente:* Konzept einer DAU-Aktivität, eine mit dieser Aktivität kompatible Systemkonfiguration

*Ausgabe:* Systemkonfiguration zur angegebenen Aktivität, wenn eine berechnet wurde. Ansonsten werden zu dem angegebenen Konzept ähnliche Aktivitäten ausgegeben, die realisiert werden können.

*Beschreibung:* Die angegebene Systemkonfiguration wird auf Kompatibilität mit der angegebenen Aktivität getestet und, wenn sie kompatibel ist, in die Datenbank ausführbarer Systemkonfigurationen aufgenommen.

#### 7.3.4.5 Methode: AddConfigurationAutomatically

*Argumente:* Konzept einer DAU-Aktivität

*Ausgabe:* Systemkonfiguration zur angegebenen Aktivität, wenn eine berechnet wurde. Ansonsten werden zu dem angegebenen Konzept ähnliche Aktivitäten ausgegeben, die realisiert werden können.

*Beschreibung:* Die automatische Konfigurationskomponente ermittelt aus der Beschreibung der bekannten DAU-Spezialisten und der angegebenen DAU-Aktivität eine konkrete Aggregation der DAU-Spezialisten (Systemkonfiguration). Der berechnete Ausführungsplan wird in die Datenbank ausführbarer Systemkonfigurationen aufgenommen.

#### 7.3.4.6 Methode: AddSubsumptionRelation

*Argumente:* Name des Oberkonzepts einer DAU-Aktivität, Name des Unterkonzepts. Beide Konzepte müssen in der Datenbank der Konzeptbeschreibungen von DAU-Aktivitäten vorkommen und es darf noch keine Subsumptionsbeziehung bestehen.

*Ausgabe:* Veränderte Hierarchie der definierten Konzeptbeschreibungen

*Beschreibung:* In die Hierarchie der definierten Konzeptbeschreibungen wird eine neue Subsumptionsbeziehung zwischen zwei Konzepten eingeführt. Mittels dieser Methode

ist es dem Systemoperator in der Phase Systemkonfiguration möglich, anwendungsspezifische Spezialisierungen generischer Konzepte von DAU-Aktivitäten einzufügen.

#### 7.3.4.7 Methode: AddPartWholeRelation

*Argumente:* Name der Zerlegungsrelation, Konzept (DAU-Aktivität) der Aggregate, Konzept der Komponenten (DAU-Aktivität), Unter- und Obergrenze der Kardinalität

*Ausgabe:* Definierte Teil/Ganzes-Beziehungen zum angegebenen Konzept der Aggregate

*Beschreibung:* Über diese Methode wird eine Paronomie über die angegebenen Konzepte definiert. Wurden unter einem Relationsnamen mehrere Zerlegungen eingegeben, so handelt es sich um alternative Zerlegungen.

#### 7.3.4.8 Methode: ModifyAttribute

*Argumente:* Name des zu modifizierenden Attributs, Name des modifizierten Konzepts einer DAU-Aktivität, Datentypbezeichnung für das Attribut

*Ausgabe:* Darstellung des modifizierten Konzepts

*Beschreibung:* Durch diese Methode kann der Systemoperator in der Konfigurationsphase den Attributvektor eines Objektes modifizieren. Kennt das Konzept den angegebenen Attributnamen nicht, so wird es neu angelegt. Ansonsten wird der Datentyp des bestehenden Attributs entsprechend verändert.

### 7.3.5 DAU-Monitor

Wesentliche Aufgabe des DAU-Monitors ist die Visualisierung des aktuellen Systemverhaltens. Dazu gehören aktive DAU-Spezialisten inklusive der Ausprägungen ihrer Parameter und ihrer Ein- bzw. Ausgabebeziehungen.

#### 7.3.5.1 Methode: ViewSystemConfiguration

*Argumente:*

*Ausgabe:* Browser-Darstellung der aktuellen Systemkonfiguration mit aktiven DAU-Spezialisten, ihren Parametern und den aktuellen Ein- bzw. Ausgabebeziehungen. Die

Anzeige umfaßt analog zum `ps`-Werkzeug des UNIX-Systems Laufzeitdaten wie aufgewendete Speicher- und Rechenzeitressourcen sowie interne Statusmeldungen der DAU-Spezialisten in der aktuellen Konfiguration.

*Beschreibung:* Diese Methode implementiert im wesentlichen die Schnittstelle Monitoring (vgl. Kapitel 6.2.2.10).

### 7.3.5.2 Methode: ChangeSystemConfiguration

*Eingabe:* Systemkonfiguration, aktive DAU-Aktivität

*Ausgabe:*

*Beschreibung:* Analog zu Kapitel 7.3.4.4 wird hier eine Systemkonfiguration festgelegt. Hier jedoch wird diese Konfiguration für die angegebene aktive DAU-Aktivität vorgeschrieben und eventuell berechnete Zwischenergebnisse werden gelöscht, sofern sie nicht in der angegebenen neuen Systemkonfiguration verwendet werden. Diese Methode implementiert die Schnittstelle Tuning (vgl. Kapitel 6.2.2.10). Zusätzlich können in dieser Schnittstelle mittels der Methode *AddConstraint* (vgl. Kapitel 7.3.4.3) neue Integritätsbedingungen in die Konzeptbeschreibungen eingeführt werden. Dadurch kann die erneute Berechnung fehlerhafter Systemkonfigurationen verhindert werden.

### 7.3.6 Verifikation und Akquisition

Diese Komponente dient sowohl der Verifikation von DAU-Ergebnissen durch den „normalen“ Benutzer als auch der Erhebung neuen Wissens – evtl. auch aufgrund von DAU-Fehlern – durch den autorisierten Benutzer. Die Daten, die durch den normalen Benutzer geändert wurden, werden gemeinsam mit ihren ursprünglichen Werten gespeichert und zu einem späteren, durch den Workflow festgelegten Zeitpunkt dem autorisierten Benutzer angezeigt. Dieser kann sich gleichzeitig über eine graphische Schnittstelle DAU-Wissen und alle Zwischenergebnisse ausgeben lassen. Dadurch kann er Fehlerquellen zurückverfolgen und zukünftig durch die Akquisition neuen Wissens ausschalten (On-Line-Teaching). Außerdem kann der autorisierte Benutzer auch neues Wissen oder die zur Akquisition von neuem Wissen benötigten Daten eingeben, ohne daß dies durch eine

vorhergehende, fehlerbehaftete Analyse motiviert ist. Hier handelt es sich dann aber eher um Massendaten als um einzelne, lokale Änderungen (Off-Line-Learning).

#### 7.3.6.1 Methode: VerifyResults

*Argumente:* Brutto-TIFF Bild, Texterkennungsergebnisse, extrahierte Information (Datenbank/Menü), Vorgangszuordnung

*Ausgabe:* Korrigierte Texterkennungsergebnisse, extrahierte Information (Datenbank/Menü), Vorgangszuordnung

*Beschreibung:* Wird durch einen Benutzer aufgerufen. Angezeigt werden nach dem Ablauf der DAU Brutto-Image, Texterkennungsergebnisse, extrahierte Information (Datenbank/Menü) und Vorgangszuordnung. Der Benutzer kann mittels einer intuitiven, einfachen Oberfläche diese Daten korrigieren. Korrigierte Daten werden in der Datenbank abgelegt. Zudem werden sie mit den fehlerbehafteten Daten zwischengespeichert, damit der autorisierte Benutzer später darauf zugreifen kann. Außerdem können sie auch direkt an Lernkomponenten lernfähiger DAU-Spezialisten weitergeleitet werden, die daraufhin ihre Wissensbasen anpassen.

#### 7.3.6.2 Methode: CheckAll

*Argumente:* Dokumentnamen

*Ausgabe:* Ursprüngliche DAU-Ergebnisse, Korrigierte DAU-Ergebnisse, DAU-Zwischenergebnisse, DAU-Wissen

*Beschreibung:* Der autorisierte Benutzer kann sich gleichzeitig über eine graphische Schnittstelle ursprüngliche DAU-Ergebnisse, korrigierte DAU-Ergebnisse, DAU-Wissen und alle Zwischenergebnisse ausgeben lassen. Dadurch kann er Fehlerquellen zurückverfolgen.

#### 7.3.6.3 Methode: AcquireLexicalKnowledge

*Argumente:* Wort, Lexikalische Eigenschaft

*Ausgabe:*

*Beschreibung:* Der autorisierte Benutzer kann hier neue Wortinformationen eingeben. Dies sind Synonym-Beziehungen und Hyper-/Hyponym-Beziehungen sowie morphologische Wortinformationen. Die Daten werden in einer Datenbank abgelegt.

#### 7.3.6.4 Methode: AcquirePatterns

*Argumente:* Eine oder mehrere ROIs, Bezeichner für ROIs, zu extrahierende Informationseinheiten innerhalb der ROIs, feste Bestandteile einer Wortsequenz

*Ausgabe:*

*Beschreibung:* Der autorisierte Benutzer kann hier Regionen kennzeichnen, in denen bestimmte Wortsequenzen (Pattern) gefunden und bestimmte Informationseinheiten extrahiert werden sollen. Die Region wird mit bisher generierten generischen Pattern verglichen. Abhängig von der Ähnlichkeit zu bestehenden Pattern wird entweder ein bestehendes erweitert oder ein neues Pattern generiert.

#### 7.3.6.5 Methode: AcquireNewClass

*Argumente:* Eine beliebige Menge von Dokumenten (TIFF-Bild oder ASCII-Text), Name für Dokumentklasse, evtl. Klassenfeatures.

*Ausgabe:*

*Beschreibung:* Der autorisierte Benutzer kann für eine beliebige Menge von Dokumenten eine neue Klasse festlegen (z. B. Nachrichtentyp, Produktkategorie, Formulartyp/-nummer). Aus den Eingabedaten wird eine Erweiterung der bestehenden Regeln für diese neue Klasse generiert.

#### 7.3.6.6 Methode: AcquireNewGraphicalDocumentFeatures

*Argumente:* Dokument als TIFF-Bild, Formular- und Tabelleneigenschaften des Dokuments in Form einer Liste mit Namen für Regions-Of-Interest (ROIs), deren geometrische Position und graphischen Eigenschaften

*Ausgabe:*

*Beschreibung:* Der autorisierte Benutzer kann die Eigenschaften, die für ein Referenzdokument für Tabellen- und Formularerkennung benötigt werden, eingeben. Dazu gehören u. a. Angaben über Linien, geometrische Positionen von ROIs, mögliche Farben des Vorder-/Hintergrunds und die Verteilung von Farben im Dokument. Diese Angaben werden gespeichert.

#### 7.3.6.7 Methode: AcquireNewData

*Argumente:* Datenbankbezeichner, Eintrag

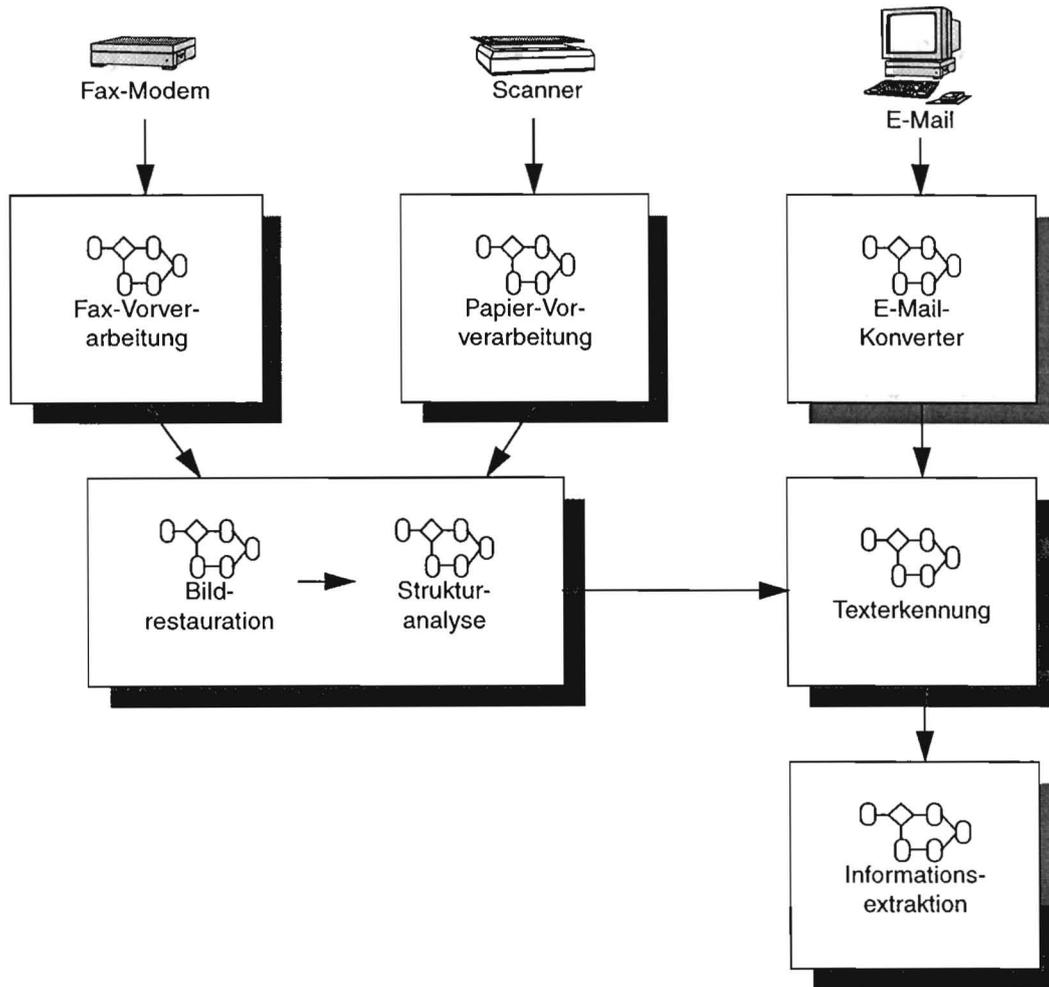
*Ausgabe:*

*Beschreibung:* Der autorisierte Benutzer kann die bestehenden, firmenspezifischen Datenbanken hier erweitern (z. B. Adressen, Leute, Firmen, Produkte). Die Daten werden in der jeweiligen Datenbank abgelegt.

### 7.3.7 DAU Spezialisten

Die Klassenkategorie „DAU Spezialisten“ dient dem Systemadministrator dazu, die innerhalb des Workflows relevanten DAU-Aktionen einzeln ansteuern zu können. Derart soll es möglich sein, z. B. eine Dokumenttyp-Klassifikation durchzuführen und danach in Abhängigkeit des Dokumenttyps verschiedene weitere Analysen des Dokumentinhalts durchzuführen. Da die einzelnen DAU-Aktionen sehr stark anwendungsspezifisch sind, sind die an dieser Stelle ausformulierten Anforderungen eher als exemplarische Minimalanforderungen für einen überzeugenden Prototypen zu verstehen. An einigen Stellen kann während der Systementwicklung eine feinerkörnigere Untergliederung sinnvoll sein, um dem Systemadministrator mehr Eingriffsmöglichkeiten auf den Trade-off zwischen Berechnungseffizienz vs. Ergebnisqualität bieten zu können. Solche Stellen sind teilweise bereits hier angegeben.

Aus Gründen der Übersichtlichkeit sind die DAU Spezialisten in zwei Gruppen eingeteilt: die erste umfaßt die Spezialisten zur *Dokumenterfassung* bis hin zum erkannten Text, die zweite Gruppe ist die der *Informationsextraktion*, welche aus Dokumentstruktur und -text gleichermaßen die interessierenden Informationen herauszieht. Abbildung 6



**ABBILDUNG 6: Komponenten der Dokumentanalyse**

gibt einen Überblick zu den Beziehungen der im folgenden beschriebenen DAU-Spezialisten.

Die bei der vorliegenden Abstraktionsstufe relevanten Schnittstellen der Dokumentanalyse sind: *Dokumentbild*, *Dokumentstruktur*, *Dokumenttext*, *Dokumentinhalt*

#### 7.3.7.1 Aufgaben zur Dokumenterfassung

- *Fax-Vorverarbeitung*: Einlesen von Faxen als *Dokumentbild*; beinhaltet Fax-spezifische Bildrestaurierung bzgl. Übertragungsfehler, Lesefehler beim Scannen im sendenden Faxgerät, u.ä.
- *Papier-Vorverarbeitung*: Einlesen von gedruckten Dokumenten als *Dokumentbild*; beinhaltet Einscannen.

- *Bildrestaurierung*: allgemeine Bildvorverarbeitung ohne Fax-spezifische Teile (Eingabe und Ausgabe sind vom Typ *Dokumentbild*); beinhaltet Ausrichtung (*de-skew*), Bildbereinigung (*cleansing*, *de-speckle*, *de-shade*), u.ä.
- *Strukturanalyse*: mit *Dokumentbild* als Eingabe und *Dokumentstruktur* als Ausgabe; beinhaltet Segmentierung und ggf. Logical Labeling oder alternativ Formularerkennung. Evtl. kann man hier eine Unterteilung in vier Komponenten vornehmen: Grobsegmentierung, Formularerkennung, Brief-Labeling, Feinsegmentierung bis auf Zeichenebene.
- *Texterkennung*: mit *Dokumentbild* und *Dokumentstruktur* als Eingabe und *Dokumenttext* als Ausgabe. Viele kommerzielle Texterkenner benötigen nur das *Dokumentbild* und erzeugen die *Dokumentstruktur* selbst, teilweise sogar ohne die Struktur als Ergebnis zu liefern. Die Texterkennung beinhaltet einen oder mehrere Einzelzeichenerkenner (OCR), einen lexikalischen Abgleich, ein Zeichen- und ein Wort-Voting für mehrere Erkenner und Verfahren zur kontextuellen Verifikation von Zeichen- und Worthypothesen.
- *E-Mail-Konverter*: Einlesen von E-Mails mit *Dokumenttext* als Eingabe und *Dokumentstruktur* und einem modifizierten *Dokumenttext* als Ausgabe. Hier werden zu einer späteren einheitlichen Weiterverarbeitung die reinen E-Mail-Texte um Layouteinheiten ergänzt. Da der *Dokumenttext* dabei auch mittels Worterkenner bearbeitet werden kann, ist der *Dokumenttext* unter Umständen danach überarbeitet. Anders betrachtet, ist der ursprüngliche *Dokumenttext* der Roh-E-Mail als Quelldatum mit dem eingescannten oder via Fax eingegangenen *Dokumentbild* von oben vergleichbar.

### 7.3.7.2 Aufgaben zur Informationsextraktion

Die Spezialisten zur Informationsextraktion erhalten generell als Eingabe *Dokumenttext* und *Dokumentstruktur* (wobei die *Dokumentstruktur* nicht zwingend genutzt werden muß), weshalb diese beiden im folgenden nicht jeweils extra genannt werden.

- *Dokumentklassifikation*: beinhaltet beliebige Techniken zur Klassifikation des Dokumentinhalts gemäß einer vorgegebenen Klassifikation (also eines Klassifikationsschemas) anhand verschiedener Eingabedaten; hierunter fallen z. B. for-

mularbasierte, layout-/fontorientierte, einzelwortbasierte (Wortfrequenzen, IR-Techniken) und wortkontextbasierte (Stereotypen, Phrasen, Text-Patterns) Verfahren und beliebige Kombinationen davon (z. B. regelbasierte, ...).

- *Produktklassifikation/Artikelextraktion*: ähnlich wie die Dokumentklassifikation kann die Produktklassifikation beliebige Techniken bzw. Eingabedaten verwenden. Über eine reine Klassifikation anhand entsprechender Kategorien hinaus, kann hier eine Informationsextraktion für die Liste der im Dokument genannten Artikel/Produkte erfolgen.
- *Absenderidentifikation*: Erkennung des Absenders des Dokuments etwa anhand der Angaben des Faxprotokolls, des E-Mail-Headers, der Formularidentifikation oder der Absenderadresse eines gedruckten Briefes. Die eigentliche Identifikation kann nur unter Bezugnahme auf eine Adreßdatenbank des *Corporate Knowledge* durchgeführt werden.
- *Belegdatenextraktion*: Bei der Belegdatenextraktion werden Belegdatum, Schreib-/Sendeort des Dokuments, Diktatzeichen etc. ermittelt.
- *Bezugsdatenextraktion*: Bei der Bezugsdatenextraktion werden Referenzen auf vorangegangene und im vorliegenden Dokument angesprochene Schreiben identifiziert. Dabei können wahlweise Dokumenttyp, Bezugsdatum, Dokumentnummer/-bezeichnung, Sachbearbeiter und Dokumentmedium (Fax, Brief, ...) anfallen.
- *Empfängeridentifikation*: anders als die Absenderidentifikation benötigt die Empfängeridentifikation möglicherweise weitere Eingabedaten, z. B. die Dokumentklasse, der Produktklasse und den Absender des Dokuments. Zwar dürfte oft eine Empfängeridentifikation anhand der Empfängeradresse oder anhand der Anrede möglich sein, aber im allgemeinen dienen zusätzliche Angaben einer Erhöhung der Sicherheit des Ergebnisses. Auch hier ist ein Zugriff auf die Mitarbeiter der Firma in einer Organisationsdatenbank des *Corporate Knowledge* sinnvoll.

Weiterhin sind eine Reihe von IE-Spezialisten denkbar und sinnvoll, die zum einen von der Dokumentstruktur abhängen (was ist auf dem Dokument geschrieben?) und zum anderen von der betrachteten Anwendung (was interessiert uns?). Stärker als bei der

Dokumenterfassung (vgl. Kapitel 7.3.7.1) ist der Entwurf einer möglichst flexiblen Toolbox von DAU-Spezialisten für die Informationsextraktion wichtig.

## **7.4 Kritikalitäten / Funktionen-Matrix**

Entfällt.



Deutsches  
Forschungszentrum  
für Künstliche  
Intelligenz GmbH

---

-Bibliothek, Information und Dokumentation (BID)- PF 2080 67608 Kaiserslautern FRG	Telefon (0631) 205 Telefax (0631) 205 e-mail dfkibib@dfki.uni WWW http://www.dfk sb.de/dfkibib
--	--

---

## Veröffentlichungen des DFKI

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder (so sie als per ftp erhaeltlich angemerkt sind) per anonymous ftp von ftp.dfki.uni-kl.de (131.246.241.100) im Verzeichnis pub/Publications bezogen werden. Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

## DFKI Publications

*The following DFKI publications or the list of all published papers so far are obtainable from the above address or (if they are marked as obtainable by ftp) by anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) in the directory pub/Publications.*

*The reports are distributed free of charge except where otherwise noted.*

---

## DFKI Research Reports

### 1997

#### RR-97-08

*Stefan Müller*

Complement Extraction Lexical Rules and Argument Attraction  
14 pages

#### RR-97-07

*Stefan Müller*

Yet Another Paper about Partial Verb Phrase Fronting in German  
26 pages

#### RR-97-06

*Stefan Müller*

Scrambling in German - Extraction into the *Mittelfeld*  
24 pages

#### RR-97-04

*Serge Autexier, Dieter Hutter*

Parameterized Abstractions used for Proof-Planning  
13 pages

#### RR-97-03

*Dieter Hutter*

Using Rippling to Prove the Termination of Algorithms  
15 pages

#### RR-97-02

*Stephan Busemann, Thierry Declerck, Abdel Kader Diagne, Luca Dini, Judith Klein, Sven Schmeier*

Natural Language Dialogue Service for Appointment Scheduling Agents  
15 pages

#### RR-97-01

*Erica Melis, Claus Sengler*

Analogy in Verification of State-Based Specifications: First Results  
12 pages

### 1996

#### RR-96-06

*Claus Sengler*

Case Studies of Non-Freely Generated Data Types  
200 pages

#### RR-96-05

*Stephan Busemann*

Best-First Surface Realization  
11 pages

#### RR-96-04

*Christoph G. Jung, Klaus Fischer, Alastair Burt*

Multi-Agent Planning  
Using an *Abductive*  
EVENT CALCULUS  
114 pages

**RR-96-03**

*Günter Neumann*  
Interleaving  
Natural Language Parsing and Generation  
Through Uniform Processing  
51 pages

**RR-96-02**

*E. André, J. Müller, T. Rist:*  
PPP-Persona: Ein objektorientierter Multimedia-Präsentationsagent  
14 Seiten

**RR-96-01**

*Claus Sengler*  
Induction on Non-Freely Generated Data Types  
188 pages

**1995****RR-95-20**

*Hans-Ulrich Krieger*  
Typed Feature Structures, Definite Equivalences,  
Greatest Model Semantics, and Nonmonotonicity  
27 pages

**RR-95-19**

*Abdel Kader Diagne, Walter Kasper, Hans-Ulrich Krieger*  
Distributed Parsing With HPSG Grammar  
20 pages

**RR-95-18**

*Hans-Ulrich Krieger, Ulrich Schäfer*  
Efficient Parameterizable Type Expansion for Typed  
Feature Formalisms  
19 pages

**RR-95-17**

*Hans-Ulrich Krieger*  
Classification and Representation of Types in TDL  
17 pages

**RR-95-16**

*Martin Müller, Tobias Van Roy*  
Title not set  
0 pages

**Note:** The author(s) were unable to deliver this document for printing before the end of the year. It will be printed next year.

**RR-95-15**

*Joachim Niehren, Tobias Van Roy*  
Title not set  
0 pages

**Note:** The author(s) were unable to deliver this document for printing before the end of the year. It will be printed next year.

**RR-95-14**

*Joachim Niehren*  
Functional Computation as Concurrent Computation  
50 pages

**RR-95-13**

*Werner Stephan, Susanne Biundo*  
Deduction-based Refinement Planning  
14 pages

**RR-95-12**

*Walter Hower, Winfried H. Graf*  
Research in Constraint-Based Layout, Visualization,  
CAD, and Related Topics: A Bibliographical Survey  
33 pages

**RR-95-11**

*Anne Kilger, Wolfgang Finkler*  
Incremental Generation for Real-Time Applications  
47 pages

**RR-95-10**

*Gert Smolka*  
The Oz Programming Model  
23 pages

**RR-95-09**

*M. Buchheit, F. M. Donini, W. Nutt, A. Schaerf*  
A Refined Architecture for Terminological Systems:  
Terminology = Schema + Views  
71 pages

**RR-95-08**

*Michael Mehl, Ralf Scheidhauer, Christian Schulte*  
An Abstract Machine for Oz  
23 pages

**RR-95-07**

*Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt*  
The Complexity of Concept Languages  
57 pages

**RR-95-06**

*Bernd Kiefer, Thomas Fettig*  
FEGRAMED  
An interactive Graphics Editor for Feature Structures  
37 pages

**RR-95-05**

*Rolf Backofen, James Rogers, K. Vijay-Shanker*  
A First-Order Axiomatization of the Theory of Finite  
Trees  
35 pages

**RR-95-04**

*M. Buchheit, H.-J. Bürckert, B. Hollunder, A. Laux, W. Nutt, M. Wójcik*  
Task Acquisition with a Description Logic Reasoner  
17 pages

**RR-95-03**

*Stephan Baumann, Michael Malburg, Hans-Guenther Hein, Rainer Hoch, Thomas Kieninger, Norbert Kuhn*  
 Document Analysis at DFKI  
 Part 2: Information Extraction  
 40 pages

**RR-95-02**

*Majdi Ben Hadj Ali, Frank Fein, Frank Hoenes, Thorsten Jaeger, Achim Weigel*  
 Document Analysis at DFKI  
 Part 1: Image Analysis and Text Recognition  
 69 pages

**RR-95-01**

*Klaus Fischer, Jörg P. Müller, Markus Fischel*  
 Cooperative Transportation Scheduling  
 an application Domain for DAI  
 31 pages

**1994****RR-94-39**

*Hans-Ulrich Krieger*  
 Typed Feature Formalisms as a Common Basis for Linguistic Specification.  
 21 pages

**RR-94-38**

*Hans Uszkoreit, Rolf Backofen, Stephan Busemann, Abdel Kader Diagne, Elizabeth A. Hinkelman, Walter Kasper, Bernd Kiefer, Hans-Ulrich Krieger, Klaus Netter, Günter Neumann, Stephan Oepen, Stephen P. Spackman.*  
 DISCO—An HPSG-based NLP System and its Application for Appointment Scheduling.  
 13 pages

**RR-94-37**

*Hans-Ulrich Krieger, Ulrich Schäfer*  
 TDL - A Type Description Language for HPSG, Part 1: Overview.  
 54 pages

**RR-94-36**

*Manfred Meyer*  
 Issues in Concurrent Knowledge Engineering. Knowledge Base and Knowledge Share Evolution.  
 17 pages

**RR-94-35**

*Rolf Backofen*  
 A Complete Axiomatization of a Theory with Feature and Arity Constraints  
 49 pages

**RR-94-34**

*Stephan Busemann, Stephan Oepen, Elizabeth A. Hinkelman, Günter Neumann, Hans Uszkoreit*  
 COSMA – Multi-Participant NL Interaction for Appointment Scheduling  
 80 pages

**RR-94-33**

*Franz Baader, Armin Laux*  
 Terminological Logics with Modal Operators  
 29 pages

**RR-94-31**

*Otto Kühn, Volker Becker, Georg Lohse, Philipp Neumann*  
 Integrated Knowledge Utilization and Evolution for the Conservation of Corporate Know-How  
 17 pages

**RR-94-23**

*Gert Smolka*  
 The Definition of Kernel Oz  
 53 pages

**RR-94-20**

*Christian Schulte, Gert Smolka, Jörg Würtz*  
 Encapsulated Search and Constraint Programming in Oz  
 21 pages

**RR-94-19**

*Rainer Hoch*  
 Using IR Techniques for Text Classification in Document Analysis  
 16 pages

**RR-94-18**

*Rolf Backofen, Ralf Treinen*  
 How to Win a Game with Features  
 18 pages

**RR-94-17**

*Georg Struth*  
 Philosophical Logics—A Survey and a Bibliography  
 58 pages

**RR-94-16**

*Gert Smolka*  
 A Foundation for Higher-order Concurrent Constraint Programming  
 26 pages

**RR-94-15**

*Winfried H. Graf, Stefan Neurohr*  
 Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces  
 20 pages

**RR-94-14**

*Harold Boley, Ulrich Buhrmann, Christof Kremer*  
 Towards a Sharable Knowledge Base on Recyclable Plastics  
 14 pages

**RR-94-13***Jana Koehler*

Planning from Second Principles—A Logic-based Approach  
49 pages

**RR-94-12***Hubert Comon, Ralf Treinen*

Ordering Constraints on Trees  
34 pages

**RR-94-11***Knut Hinkelmann*

A Consequence Finding Approach for Feature Recognition in CAPP  
18 pages

**RR-94-10***Knut Hinkelmann, Helge Hintze*

Computing Cost Estimates for Proof Strategies  
22 pages

**RR-94-08***Otto Kühn, Björn Höfling*

Conserving Corporate Knowledge for Crankshaft Design  
17 pages

**RR-94-07***Harold Boley*

Finite Domains and Exclusions as First-Class Citizens  
25 pages

**RR-94-06***Dietmar Dengler*

An Adaptive Deductive Planning System  
17 pages

**RR-94-05***Franz Schmalhofer, J. Stuart Aitken, Lyle E. Bourne jr.*

Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse  
81 pages

**RR-94-03***Gert Smolka*

A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards  
34 pages

**RR-94-02***Elisabeth André, Thomas Rist*

Von Textgeneratoren zu Intellimedia-Präsentationssystemen  
22 Seiten

**RR-94-01***Elisabeth André, Thomas Rist*

Multimedia Presentations: The Support of Passive and Active Viewing  
15 pages

**DFKI Technical Memos****1997****TM-97-01***Markus Perling*

GeneTS: A Relational-Functional Genetic Algorithm for the Traveling Salesman Problem  
26 pages

**1996****TM-96-02***Harold Boley*

Knowledge Bases in the World Wide Web: A Challenge for Logic Programming  
8 pages

**TM-96-01***Gerd Kamp, Holger Wache*

CTL — a description Logic with expressive concrete domains  
19 pages

**1995****TM-95-04***Klaus Schmid*

Creative Problem Solving and Automated Discovery — An Analysis of Psychological and AI Research --  
152 pages

**TM-95-03***Andreas Abecker, Harold Boley, Knut Hinkelmann, Holger Wache,**Franz Schmalhofer*

An Environment for Exploring and Validating Declarative Knowledge  
11 pages

**TM-95-02***Michael Sintek*

FLIP: Functional-plus-Logic Programming on an Integrated Platform  
106 pages

**TM-95-01***Martin Buchheit, Rüdiger Klein, Werner Nutt*

Constructive Problem Solving: A Model Construction Approach towards Configuration  
34 pages

1994

**TM-94-05**

*Klaus Fischer, Jörg P. Müller, Markus Pischel*  
Unifying Control in a Layered Agent Architecture  
27 pages

**TM-94-04**

*Cornelia Fischer*  
PAntUDE – An Anti-Unification Algorithm for Expressing Refined Generalizations  
22 pages

**TM-94-03**

*Victoria Hall*  
Uncertainty-Valued Horn Clauses  
31 pages

**TM-94-02**

*Rainer Bleisinger, Berthold Kröll*  
Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations  
11 pages

**TM-94-01**

*Rainer Bleisinger, Klaus-Peter Gores*  
Text Skimming as a Part in Paper Document Understanding  
14 pages

---

## DFKI Documents

1997

**D-97-06**

*Tilman Becker, Stephan Busemann, Wolfgang Finkler*  
DFKI Workshop on Natural Language Generation  
67 pages

**D-97-05**

*Stephan Baumann, Majdi Ben Hadj Ali, Jürgen Lichter, Michael Malburg, Harald Meyer auf'm Hofe, Claudia Wenzel*  
Anforderungen an ein System zur Dokumentanalyse im Unternehmenskontext  
— Integration von Datenbeständen, Aufbau- und Ablauforganisation  
42 Seiten

**D-97-04**

*Claudia Wenzel, Markus Junker*  
Entwurf einer Patternbeschreibungssprache für die Informationsextraktion in der Dokumentanalyse  
24 Seiten

**D-97-03**

*Andreas Abecker, Stefan Decker, Knut Hinkelmann, Ulrich Reimer*  
Proceedings of the Workshop „Knowledge-Based Systems for Knowledge Management in Enterprises“ 97  
167 pages

**D-97-02**

*Tilman Becker, Hans-Ulrich Krieger*  
Proceedings of the Fifth Meeting on Mathematics of Language (MOL5)  
168 pages

**D-97-01**

*Thomas Malik*  
NetGLTool Benutzeranleitung  
40 Seiten

1996

**D-96-07**

*Technical Staff*  
DFKI Jahresbericht 1995  
55 Seiten

**Note:** This document is no longer available in printed form.

**D-96-06**

*Klaus Fischer (Ed.)*  
Working Notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems  
63 pages

**D-96-05**

*Martin Schaaf*  
Ein Framework zur Erstellung verteilter Anwendungen  
94 pages

**D-96-04**

*Franz Baader, Hans-Jürgen Bürckert, Andreas Günter, Werner Nutt (Hrsg.)*  
Proceedings of the Workshop on Knowledge Representation and Configuration WRKP'96  
83 pages

**D-96-03**

*Winfried Tautges*  
Der DESIGN-ANALYZER - Decision Support im Designprozess  
75 Seiten

**D-96-01**

*Klaus Fischer, Darius Schier*  
Ein Multiagentenansatz zum Lösen von Fleet-Scheduling-Problemen  
72 Seiten

## 1995

### D-95-12

*F. Baader, M. Buchheit, M. A. Jeusfeld, W. Nutt (Eds.)*  
Working Notes of the KI'95 Workshop:  
KRDB-95 - Reasoning about Structured Objects:  
Knowledge Representation Meets Databases  
61 pages

### D-95-11

*Stephan Busemann, Iris Merget*  
Eine Untersuchung kommerzieller Terminverwaltungssoftware im Hinblick auf die Kopplung mit natürlich-sprachlichen Systemen  
32 Seiten

### D-95-10

*Volker Ehresmann*  
Integration ressourcen-orientierter Techniken in das wissensbasierte Konfigurierungssystem TOOCON  
108 Seiten

### D-95-09

*Antonio Krüger*  
PROXIMA: Ein System zur Generierung graphischer Abstraktionen  
120 Seiten

### D-95-08

*Technical Staff*  
DFKI Jahresbericht 1994  
63 Seiten

**Note:** This document is no longer available in printed form.

### D-95-07

*Ottmar Lutzy*  
Morphic - Plus  
Ein morphologisches Analyseprogramm für die deutsche Flexionsmorphologie und Komposita-Analyse  
74 Seiten

### D-95-06

*Markus Steffens, Ansgar Bernardi*  
Integriertes Produktmodell für Behälter aus Faserverbundwerkstoffen  
48 Seiten

### D-95-05

*Georg Schneider*  
Eine Werkbank zur Erzeugung von 3D-Illustrationen  
157 Seiten

### D-95-04

*Victoria Hall*  
Integration von Sorten als ausgezeichnete taxonomische Prädikate in eine relational-funktionale Sprache  
56 Seiten

### D-95-03

*Christoph Endres, Lars Klein, Markus Meyer*  
Implementierung und Erweiterung der Sprache *ALCP*  
110 Seiten

### D-95-02

*Andreas Butz*  
BETTY  
Ein System zur Planung und Generierung informativer Animationssequenzen  
95 Seiten

### D-95-01

*Susanne Biundo, Wolfgang Tank (Hrsg.)*  
PuK-95, Beiträge zum 9. Workshop „Planen und Konfigurieren“, Februar 1995  
169 Seiten

**Note:** This document is available for a nominal charge of 25 DM (or 15 US-\$).

## 1994

### D-94-15

*Stephan Oepen*  
German Nominal Syntax in HPSG  
— On Syntactic Categories and Syntagmatic Relations  
—  
80 pages

### D-94-14

*Hans-Ulrich Krieger, Ulrich Schäfer*  
TDL - A Type Description Language for HPSG, Part 2: User Guide.  
72 pages

### D-94-12

*Arthur Sehn, Serge Autexier (Hrsg.)*  
Proceedings des Studentenprogramms der 18. Deutschen Jahrestagung für Künstliche Intelligenz KI-94  
69 Seiten

### D-94-11

*F. Baader, M. Buchheit, M. A. Jeusfeld, W. Nutt (Eds.)*  
Working Notes of the KI'94 Workshop: KRDB'94 - Reasoning about Structured Objects: Knowledge Representation Meets Databases  
65 pages

**Note:** This document is no longer available in printed form.

### D-94-10

*F. Baader, M. Lenzerini, W. Nutt, P. F. Patel-Schneider (Eds.)*  
Working Notes of the 1994 International Workshop on Description Logics  
118 pages

**Note:** This document is available for a nominal charge of 25 DM (or 15 US-\$).

### D-94-09

*Technical Staff*  
DFKI Wissenschaftlich-Technischer Jahresbericht 1993  
145 Seiten

**D-94-08***Harald Feibel*

IGLOO 1.0 - Eine grafikunterstützte Beweisentwicklungsumgebung  
58 Seiten

**D-94-07***Claudia Wenzel, Rainer Hoch*

Eine Übersicht über Information Retrieval (IR) und NLP-Verfahren zur Klassifikation von Texten  
25 Seiten

**Note:** This document is no longer available in printed form.

**D-94-06***Ulrich Buhrmann*

Erstellung einer deklarativen Wissensbasis über recyclingrelevante Materialien  
117 Seiten

**D-94-04***Franz Schmalhofer, Ludger van Elst*

Entwicklung von Expertensystemen: Prototypen, Tiefenmodellierung und kooperative Wissensentwicklung  
22 Seiten

**D-94-03***Franz Schmalhofer*

Maschinelles Lernen: Eine kognitionswissenschaftliche Betrachtung  
54 Seiten

**Note:** This document is no longer available in printed form.

**D-94-02***Markus Steffens*

Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff  
90 pages

**D-94-01***Josua Boon (Ed.)*

DFKI-Publications: The First Four Years  
1990 - 1993  
75 pages

**Anforderungen an ein System zur Dokumentanalyse im Unternehmenskontext**

**D-97-05**  
Document

**—  
Integration von Datenbeständen, Aufbau- und Ablauforganisation**

**Stephan Baumann, Majdi Ben Hadj Ali, Jürgen Lichter, Michael Malburg, Harald Meyer auf'm Hofe, Claudia Wenzel**