Reihe: Telekommunikation @ Mediendienste · Band 10

Herausgegeben von Norbert Szyperski, Udo Winand, Dietrich Seibt, Rainer Kuhlen, Rudolf Pospischil und Claudia Löbbecke

Martin Engelien/Detlef Neumann (Hrsg.)

Virtuelle Organisation und Neue Medien 2000

Workshop GeNeMe2000 Gemeinschaften in Neuen Medien

TU Dresden, 5. und 6. Oktober 2000

JOSEF EUL VERLAG
Lohmar · Köln

Reihe: Telekommunikation @ Mediendienste · Band 10

Herausgegeben von Prof. Dr. Dr. h. c. Norbert Szyperski, Köln, Prof. Dr. Udo Winand, Kassel, Prof. Dr. Dietrich Seibt, Köln, Prof. Dr. Rainer Kuhlen, Konstanz, Dr. Rudolf Pospischil, Brüssel, und Prof. Dr. Claudia Löbbecke, Köln

PD Dr.-Ing. habil. Martin Engelien
Dipl.-Inf. Detlef Neumann (Hrsg.)

Virtuelle Organisation und Neue Medien 2000

Workshop GeNeMe2000
Gemeinschaften in Neuen Medien

TU Dresden, 5. und 6. Oktober 2000

JOSEF EUL VERLAG
Lohmar · Köln

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

GeNeMe <2000 Dresden>:
GeNeMe 2000 : Gemeinschaften in neuen Medien ; Dresden, 5. und 6. Oktober 2000, an der Fakultät Informatik an der Technischen Universität Dresden / Technische Universität Dresden, Fakultät Informatik, Institut für Angewandte Informatik, Privat-Dozentur „Angewandte Informatik". Martin Engelien ; Detlef Neumann (Hrsg.). – Lohmar ; Köln : Eul, 2000
(Reihe: Telekommunikation und Mediendienste ; Bd. 10)
ISBN 3-89012-786-X

Josef Eul Verlag GmbH
Brandsberg 6
53797 Lohmar
Tel.: 0 22 05 / 91 08 91
Fax: 0 22 05 / 91 08 92
http://www.eul-verlag.de
info@eul-verlag.de

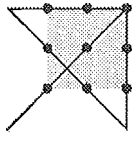
Printed in Germany
Druck: Rosch-Buch, Scheßlitz

Technische Universität Dresden
Fakultät Informatik • Institut für Angewandte Informatik
Privat-Dozentur „Angewandte Informatik“

PD Dr.-Ing. habil. Martin Engelien
Dipl.-Inf. Detlef Neumann
(Hrsg.)

an der
Fakultät Informatik der Technischen Universität Dresden

gefördert von der Klaus Tschira Stiftung,
gemeinnützige Gesellschaft mit beschränkter Haftung,
unter Mitwirkung der Gesellschaft für Informatik e.V., Regionalgruppe Dresden



am 5. und 6. Oktober 2000
in Dresden

http://www-emw.inf.tu-dresden.de/geneme
Kontakt: Detlef Neumann (dn3@inf.tu-dresden.de)

F.3. Eine virtuelle Gemeinschaft für die Planung von Servicerobotern

O. Taminé
Prof. Dr. R. Dillmann
Institut für Prozessrechentechnik, Universität Karlsruhe

1. Einleitung

Die Forderungen des Marktes nach immer leistungsfähigeren Produkten resultiert zum einen in einer stetig steigenden Variantenvielfalt, zum anderen in einem Anstieg der Produktkomplexität. Steigende Produktfunktionalität und -komplexität, immer kürzer werdende Innovationszyklen bei gleichzeitiger Steigerung der Qualität und Reduzierung von Kosten, stellen immer höhere Anforderungen an die Entwickler und an die Struktur des Produktentwicklungsprozesses.

Die Folge ist eine steigende Anzahl eingebundener Experten aus unterschiedlichen natur- und ingenieurwissenschaftlichen Disziplinen sowie die Notwendigkeit einer detaillierten Produktplanung. Dies erfordert eine enge Einbindung aller an der Produktentwicklung beteiligten Personen. Insbesondere sind dies der Kunde, der seine Wünsche äußert, die Entwickler, die das technische Know-How besitzen und Projektpartner aus unterschiedlichen Industriebranchen, die auf die Entwicklung Einfluß nehmen.

2. Motivation und Problemstellung

Die Entwicklung von Servicerobotern ist geprägt durch deren vielfältige Einsatzbereiche. Beispielsweise werden Serviceroboter zum Auftanken von Kraftfahrzeugen, für den Bau von Hochhäusern, zur Gebäudereinigung, im Hotelbereich und im Weltraum eingesetzt [ScSc98]. Durch diese sehr unterschiedlichen Aufgabenstellungen gibt es ein breites Variantenspektrum von unterschiedlichen Typen von Servicerobotern. Somit gibt es zur Zeit keinen Standardserviceroboter, der sich automatisiert produzieren läßt und an die jeweilige Aufgabe angepaßt werden kann. Statt dessen ist für jeden Aufgabenzweck eine Spezialanfertigung notwendig.

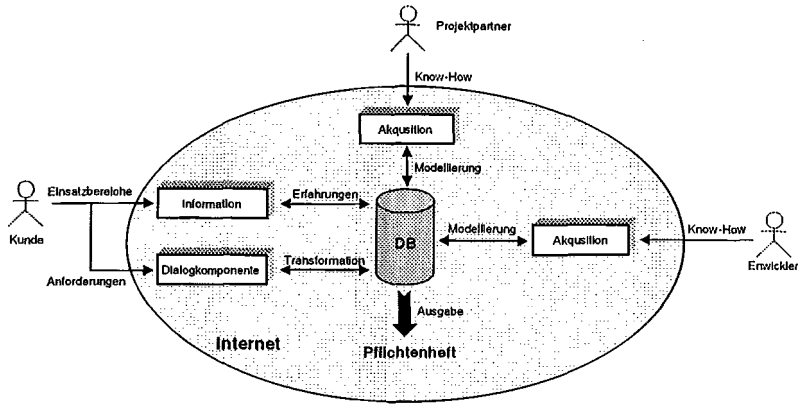


Abbildung 1: Systemaufbau für die interdisziplinäre Serviceroboterplanung

Der Bereich der Serviceroboter stellt einen sehr jungen Forschungsbereich dar, der ständig dem Wandel des schnellen technischen Fortschritts unterliegt. Dies erschwert die Planung, da sich die Konstruktionsprozesse oft nicht wiederholen und eine Automatisierung dieser Vorgänge somit nur schwer durchführbar ist [TaDi99].

Aus diesen Gründen wird ein System benötigt, welches Kunden und Experten den dynamischen Wissensaustausch ermöglicht (s. Abbildung 1). Dabei wird das Ziel verfolgt, mit diesen Informationen anhand der Kundenwünsche ein Pflichtenheft zu erstellen. Dieses bildet die Grundlage für die spätere Konstruktion. Hierbei fällt dem Pflichtenheft die Aufgabe zu, zu beschreiben, welche Komponenten hierfür notwendig sind, welche Komponenten wiederverwertet werden können, welche Änderungen vorgenommen werden müssen und welche komplett neu erstellt werden müssen.

3. Systemaufbau

Dem Stellenwert der Ressource „Information" fällt heutzutage eine immer größere Bedeutung zu. Dies ist zum einem daran ersichtlich, daß immer mehr Unternehmen ihre Informationsstrukturen um Data Warehouses ergänzen [Inmo96] und zum anderen an dem starken Zuwachs von IT-Software in den Unternehmen. So wuchs der Markt für Dokumentenmanagementsysteme im Jahr 1997 um 24% [BAP98]. Das erste Ziel des im folgenden vorgestellten Systems ist es, das verteilte Kooperationswissen effizienter zu erfassen, zu organisieren und zugänglich zu machen. Hierzu ist es zunächst erforderlich, dieses Wissen genauer zu spezifizieren.

3.1 Charakterisierung von Wissen

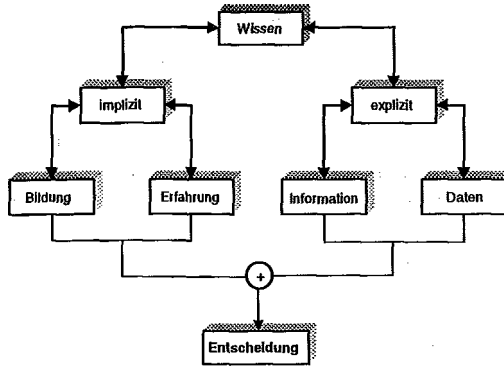


Abbildung 2: Charakterisierung von Wissen

Die in [NoTa97] beschriebene grundlegende Charakterisierung von Wissen unterscheidet zwischen explizitem und implizitem Wissen (s. Abbildung 2). Explizites Wissen basiert auf Informationen und Daten. Es ist objektiver Natur und enthält generalisierende Aspekte, die nicht nur einzelnen Individuen dienen. Für die Erfassung, Verwaltung und Suche von explizitem Wissen werden heutzutage meist Dokumentenverwaltungs- und Groupwaresysteme eingesetzt [Jabo97]. Für die Analyse großer expliziter Datenbestände sind Data Mining Techniken sowie Filter- und Agentensysteme sehr beliebt [CaHa98]. Zusammenfassend kann festgestellt werden, daß für die Erfassung und Verarbeitung von explizitem Wissen bereits eine Reihe softwaretechnischer Lösungen gefunden wurde.

Anders sieht es im Bereich des impliziten Wissens aus, welches auf Erfahrungen einzelner basiert, sehr subjektiv und in der Regel nicht generalisierbar ist. Implizites Wissen wird sehr oft unbewußt angewandt, wenn auf Grund von bereits gewonnenen Erfahrungen oder erlerntem Wissen Entscheidungen getroffen werden. So führen Personen bestimmte Aktivitäten aus, ohne daß sie formulieren könnten, welche Schritte unter welchen Bedingungen ihren Handlungen zu Grunde liegen. Auch im Alltag ist dieses Phänomen wohlbekannt: Wir sind in der Lage, in unserer Muttersprache grammatisch korrekte Sätze zu bilden, ohne die Regeln dieser Grammatik angeben zu können. Deshalb wird implizites Wissen oftmals nicht erfaßt und es gibt nur sehr wenig Softwareunterstützung.

Bezeichnung	Verkörperung	Form	Beispiele
Abstrakt, weich	Struktur, Kultur, Rollen	Implizit	Organisationskultur, informelle Machtstrukturen
Semi-abstrakt	Geschichten, Mythen	Implizit	Anekdoten, Ereignisse, Kundenberichte
Semi-konkret	Technisches Know-How, Standardprozeduren, Geschäftsregeln	Implizit + Explizit	Problembeschreibungen, Lösungsbeschreibungen, Geschäftsprozeßdefinitionen
Konkret, hart	Daten	Explizit	Stammdaten, Verkaufszahlen, Technische Normen, Spezifikationen

Tabelle 1: Wissenscharakterisierung nach [Warg97]

Sehr bedeutungsvoll für den Konstruktionsprozeß ist die Verknüpfung von explizitem und implizitem Wissen, da beide Wissensformen in diesen Prozeß einfließen (s. Tabelle 1). So stellt das explizite Wissen die Grundlage des Konstruierens dar, während das implizite Wissen aus früheren Entwicklungen die Qualität des neuen Entwurfs maßgeblich beeinflußt. Der soeben beschriebene Sachverhalt soll am Beispiel eines Antriebs für einen mobilen Roboter beschrieben werden. Die Unterscheidung zwischen Differentialantrieb, Vierradantrieb und Mecanumantrieb bezüglich Kosten, Bewegungsspektrum, Handhabung und Anzahl der Motoren stellt explizites Wissen dar. Implizites Wissen ist, welcher Antrieb sich innerhalb bzw. außerhalb eines Gebäudes bewährt hat oder welcher Antrieb sich besonders gut bzw. schlecht mit welcher Steuerungssoftware verträgt. Sinnvolle Konstruktionsentscheidungen können jedoch nur mit der Verknüpfung von impliziten und explizitem Wissen getroffen werden. Softwaretechnisch ist die gemeinsame Erfassung von implizitem und explizitem Wissen sowie ihrer Verknüpfung noch nicht zufriedenstellend gelöst.

3.2 Dynamisches Datenmodell

Die zentrale Komponente des in Abbildung 1 dargestellten Systems ist eine objektorientierte Datenbank. Sie modelliert das von den Projektpartnern und Entwicklern eingegebene Wissen. Dem Kunden stellt sie eine Beschreibung von früheren Projekten zur Verfügung, um ihm Einsatzmöglichkeiten aufzuzeigen. Mittels der Dialogkomponente vermittelt der Kunde dem System, welche Anforderungen er an das zu entwickelnde Produkt stellt.

Aufbauend auf dem in [EsJe90] entwickelten zweidimensionalen objektorientierten Repräsentationsmodell basiert dieses System auf einem dreidimensionalen Wissensmodell. Das zweidimensionale Modell verfügt über horizontale und vertikale Verknüpf-

ungen. Dabei stellt eine horizontale Verknüpfung eine „hat-Beziehung" dar und eine vertikale Verknüpfung eine „ist-Beziehung". Eine „hat-Beziehung" umschreibt, daß ein Objekt ein anderes Objekt beinhaltet. Eine „ist-Beziehung" signalisiert, daß ein Objekt eine Verfeinerung eines übergeordneten Objektes darstellt. Diese beiden Verknüpfungsarten reichen jedoch nicht aus, um das Konstruktionswissen für mobile Roboter vollständig abzubilden, da sich nicht alle Konstruktionsbeziehungen von Roboterkomponenten durch eine „hat"- bzw. „ist"-Beziehung modellieren lassen. Aus diesem Grund führt das vorgestellte System durch den Aufbau von Ebenen eine dritte Dimension ein.

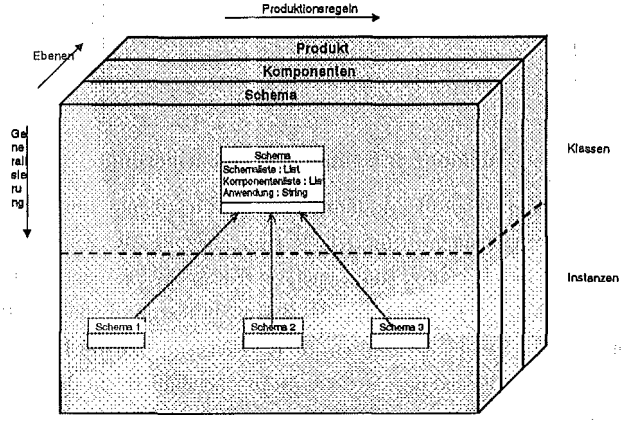


Abbildung 3: 3D-Repräsentationsmodell

Das objektorientierte Datenmodell basiert auf einem dreidimensionalen Repräsentationsmodell (s. Abbildung 3). Die Schema-Ebene beschreibt die Planungsprozesse, deren Basis die gewünschte Funktionalität des Roboters bildet. Ein Schema enthält alle hierfür notwendigen Komponenten bzw. verweist auf ein anderes Schema, in dem diese Komponenten genauer beschrieben sind. Die Komponenten-Ebene beschreibt die Funktionalitäten der einzelnen Komponenten. Dabei dürfen Komponenten aus weiteren Unter-Komponenten bestehen. Zwischen den Komponenten können ODER-Beziehungen oder UND-Beziehungen bestehen. Dabei stellt eine ODER-Beziehung eine Auswahlmöglichkeit zwischen den Komponenten dar und eine UND-Beziehung beschreibt, daß zur Realisierung beide Komponenten benötigt werden. Die Produkt-Ebene beschreibt die Produkte verschiedener Hersteller einer spezifischen Komponente. Hierbei können zwischen Produkten nur ODER-Beziehungen auftreten, da alle eventuellen UND-Beziehungen schon auf Komponenten-Ebene realisiert werden.

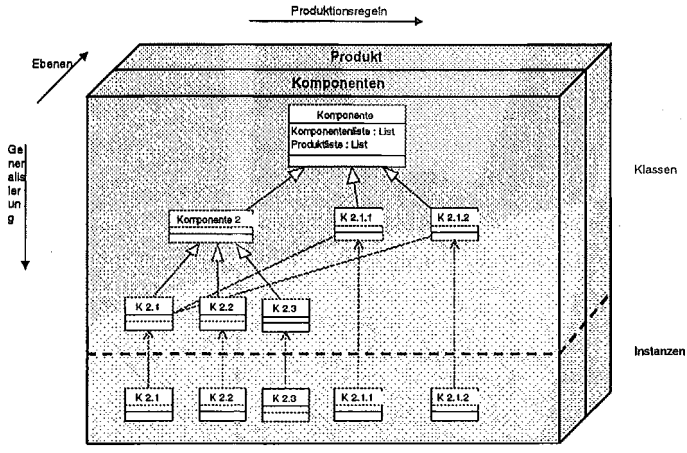


Abbildung 4: Komponentenmodellierung

Abbildung 4 zeigt beispielhaft die Modellierung der Komponente 2 in dem System. Die Basisklasse aller Komponenten ist die Klasse Komponente. Alle weiteren Komponentenklassen sind von ihr abgeleitet, somit auch die Klasse Komponente 2. Diese Klasse ist Vaterklasse von K2.1, K2.2 und K2.3 und beschreibt hierdurch, daß zur Realisierung von Komponente 2 eine Wahl einer dieser drei Klassen erforderlich ist. Entscheidet man sich für K2.1 wird dort anhand des Attributs Komponentenliste auf die Klassen K2.1.1 und K2.1.2 verwiesen. Dies bedeutet inhaltlich, daß beide Klassen zur Realisierung von K2.1 erforderlich sind. Da die Komponenten K2.1.1 und K2.1.2 nicht von K2.1 abgeleitet sind, besitzen diese als Vaterklasse direkt die Basisklasse Komponente. Umgekehrt wäre eine Ableitung von K2.1 ja gleichbedeutend mit einer ODER-Auswahl zwischen K2.1.1 und K2.1.2, was im hier vorliegenden Falle falsch wäre.

3.3 Akquisition

Als Wissensakquisition wird der Vorgang des Wissenserwerbs bezeichnet. Das Anwendungswissen muß den Wissensquellen entnommen, transformiert und in die Wissensbasis des Systems übertragen werden. Dabei erwirbt das vorgestellte System das Wissen von seinen menschlichen Experten (den Entwicklern).

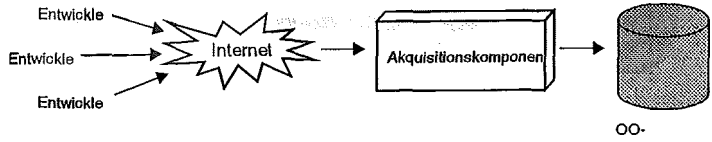


Abbildung 5: Direkte Wissensakquisition

Zur Akquisition gibt es drei verschiedene Methoden. Die indirekte Wissensakquisition liegt vor, wenn zwischen dem Experten und der Datenbank ein Administrator zwischengeschaltet wird. Der Vorteil dieser Methode liegt darin, daß nur der Administrator das System bedienen können muß. Nachteile sind der zusätzliche Aufwand (Der Experte muß das Wissen zunächst dem Administrator mitteilen, dann muß dieser es in das System einfügen) und Interpretationsprobleme, falls der Administrator den Experten nicht korrekt wiedergibt und dem System somit nicht das vollständige Wissen vermittelt. Bei der Methode der direkten Wissensakquisition kommuniziert der Experte mit dem System direkt. Voraussetzung für diese Methode ist eine mächtige Akquisitionskomponente, die dem Experten eine einfache Eingabe seines Wissens ermöglicht. Die dritte Methode ist die automatische Wissensakquisition. Hierbei erweitert das System seine Wissensbasis durch den Einsatz von Lernverfahren selbständig. Da diese Form zum Aufbau einer neuen Wissensbasis nicht eingesetzt werden kann, wird sie im weiteren Verlauf nicht näher betrachtet. Das System verwendet aus zwei Gründen das Verfahren der direkten Wissensakquisition an (s. Abbildung 5). Zum einen wird der beträchtliche Zusatzaufwand der indirekten Methode eingespart und zum anderen besitzen Entwickler von Servicerobotern im allgemeinen gute Informatikkenntnisse, die eine schnelle Einarbeitung und ein ausreichendes Verständnis zur Wissenseingabe ermöglichen.

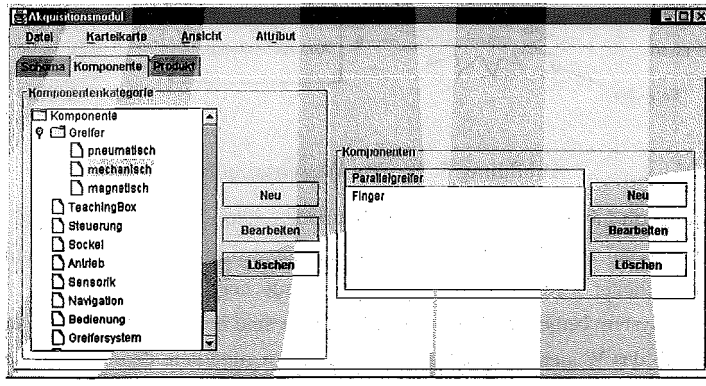


Abbildung 6: Das Akquisitionsmodul

Die Akquisitionskomponente des Systems ist für die Erfassung aller Objekte und deren Beziehungen untereinander (UND-/ODER-Verknüpfungen) auf allen drei Ebenen zuständig. Sie fügt den Objekten ihre Informationen hinzu, bestimmt die Attribute jeder Klasse und definiert für jedes Attribut den zugehörigen Wertebereich (s. Abbildung 6).

3.4 Dialogkomponente

Die Aufgabe der Dialogkomponente ist es, den Dialog zwischen dem Benutzer und dem System zu steuern. Im wesentlichen gilt es dabei, das Wissen aus der systeminternen Darstellung so aufzubereiten, daß es für den Benutzer verständlich ist und die Kommunikation in einer benutzeradäquaten Form durchgeführt werden kann.

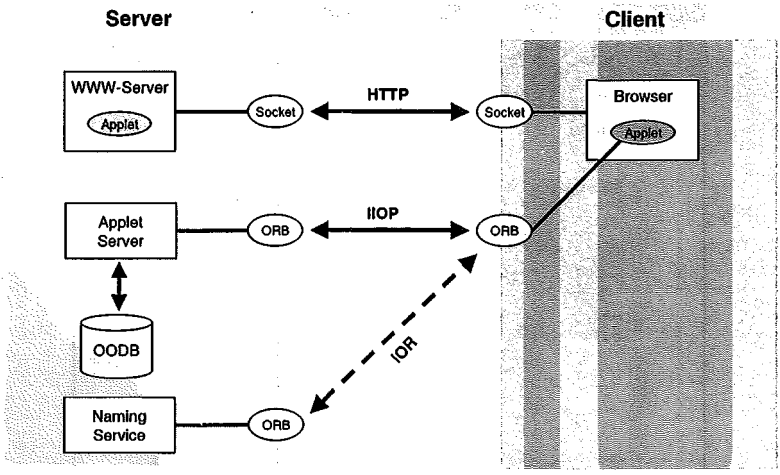


Abbildung 7: Kommunikationsablauf der Dialogkomponente

Abbildung 7 stellt den Ablauf einer Dialogsitzung dar. Die Dialogkomponente wurde als Applet realisiert und liegt auf dem WWW-Server bereit. Dieses wird mittels dem HTTP-Protokoll an den Client-Browser übertragen. Dort erzeugt das Applet einen CORBA-ORB und kontaktiert den Naming Service des Servers. Dieser liefert dem Applet die IOR des Applet-Servers, welcher für den Zugriff auf die objektorientierte Datenbank zuständig ist. Mittels IOR greift das Applet über das IIOP-Protokoll auf die CORBA-Objekte des Applet Server zu. Die CORBA-Objekte des Servers stellen sämtliche Funktionen zum Zugriff, zur Auswertung und zur Übertragung der Datenbankobjekte zur Verfügung.

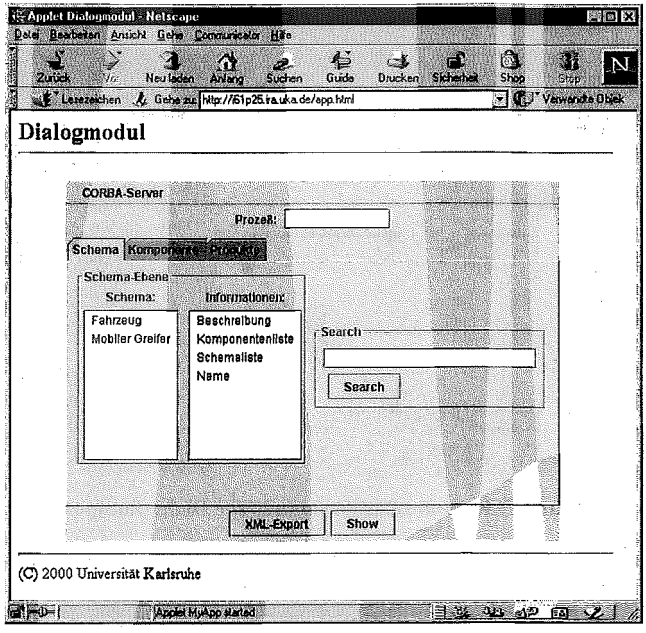


Abbildung 8: Das Applet „Dialogmodul"

Das Dialogmodul ist ähnlich dem in Abbildung 6 dargestellten Akquisitionsmodul aufgebaut. Im Unterschied zu diesem bietet es jedoch keine Funktionen zum Hinzufügen bzw. Ändern von Objekten, sondern dient lediglich zur Darstellung dieser. Das Applet ist in drei Bereiche eingeteilt (s. Abbildung 8). Jedes Tableau ist für eine der Ebenen des Datenmodells (Schema, Komponente, Produkt) zuständig. Dabei befindet sich im linken Auswahlfeld eine Liste aller Objekte einer Ebene, während rechts eine Liste der Attribute der Objekttypen zur Auswahl aufgeführt ist. Somit wird im linken Feld ausgewählt, von welchem Objekt man Informationen möchte und rechts erfolgt die Wahl welche Informationen man möchte. Die Informationen können entweder direkt angezeigt oder in eine XML-Datei expotiert werden (s. Abschnitt 3.5).

3.5 Externe Schnittstellen

Das System wurde so konzipiert, daß es mit anderen Groupware Tools zusammen eingesetzt werden kann [TaDi99]. Das zum Austausch verwendete Datenformat wurde mittels XML spezifiert und kann über eine Middleware-Architektur ausgetauscht werden [TaDi00].

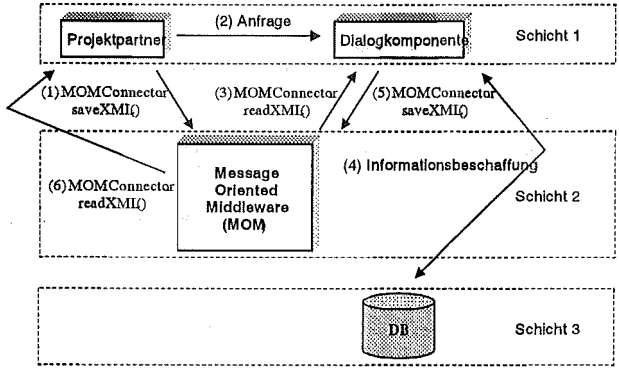


Abbildung 9: Middleware-Architektur für den XML-Datenaustausch

Die Architektur besteht aus insgesamt drei Schichten (s. Abbildung 9). Zentrales Element ist die Message Oriented Middleware (MOM). Sie bewerkstelligt den Austausch der XML-Dateien, indem sie das Aufspielen, Speichern und Herunterladen der Dateien unterstützt. Dabei interagiert der Benutzer lediglich mit der Dialogkomponente auf Schicht 1. Sie wickelt sowohl die Informationsbeschaffung aus der Datenbank sowie die MOM-Kommunikation für den Benutzer unsichtbar im Hintergrund ab. Zur Kommunikation mit der MOM bedient sie sich dabei der Klasse MOMConnector. Diese Klasse stellt Methoden zum Verbindungsauf-/abbau zur MOM, sowie zum Lesen und Schreiben von XML-Dateien zur Verfügung.

4. Zusammenfassung

Es wurde ein System zur Unterstützung einer virtuellen Gemeinschaft für die Planung von Servicerobotern vorgestellt. Durch die vielfätigen Einsatzbereiche und den schnellen technischen Fortschritt bei Servicerobotern ist das primäre Ziel des Systems der dynamische Wissensaustauch zwischen Kunden, Experten und anderen Projektteilnehmern.

Zur Wissensverarbeitung wurde ein dreidimensionales, objektorientiertes Datenmodell entwickelt, welches in die drei Ebenen Schema, Komponente und Produkt untereilt ist. Zur Eingabe des Wissens wurde ein Akqusitionsmodul entworfen. Das Applet „Dialogkomponente" ermöglicht die Navigation innerhalb des objektorientierten Wissensmodell. Hierbei kommuniziert das Modul mittels CORBA mit einem Applet Server, der die Datenbankkommunikation abwickelt. Weiterhin wurden die Schnittstellen vorgestellt, mit denen das System in andere Umgebungen integriert werden kann.

5. Literatur

[BAP98] Bullinger, H.-J.; Altenhofen,C.; Petrovic, M.: Marktstudie Dokumenten- und Workflow-Management-Systeme, Fraunhofer IAO, Fraunhofer IRB Verlag, Stuttgart, 1998.

[CaHa98] Caglayan, A.K.; Harrison, C.: Intelligente Software-Agenten: Grundlagen, Technik und praktische Anwendungen in Unternehmen; Hanser Verlag; München; 1998.

[EsJe90] Escamilla, J.; Jean, P.: Relationships in an Object Knowledge Representation Model; In Proceedings of 2nd International IEEE Conference on Tools for Artificial Intelligence, Herndon, 1990.

[Inmo96] Inmon, W.H.: Building the Data Warehouse; QED Publishing Group, New York; 1996.

[Jabo97] Jablonski, S.: Workflow-Management: Entwicklung von Anwendungen und Systemen; dpunkt-Verlag, Heidelberg; 1997.

[NoTa97] Nonaka, I.; Takeuchi, H.: Die Organisation des Wissens: Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen; Campus Verlag, Frankfurt; 1997.

[ScSc98] Schraft, R.-D.; Schmierer, G.: Serviceroboter – Produkte, Szenarien, Visionen; Springer Verlag, Berlin, Heidelberg; 1998.

[TaDi99] Taminé, O.; Dillmann, R.: A Virtual Roundtable for Interdisciplinary and Distributed Planning Processes; In Proceedings of 4th International Workshop on CSCW in Design, Compiègne, 1999.

[TaDi00] Taminé, O.; Dillmann, R.: Eine Kommunikationsarchitektur für den Wissensaustausch in interdisziplinären Projekten; in: VDI-Fortschritt-Berichte, 12. Forum Bauinformatik, Berlin, 2000.

[Warg97] Wargitsch, C.: Ein Organizational-Memory-basierter Ansatz für ein lernendes Workflow-Management-System; Arbeitsbericht des Bayrischen Forschungszentrums für Wissensbasierte Systeme, FR-1997-004; 1997.