

Reihe: Telekommunikation @ Mediendienste · Band 14

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, Prof. Dr. Claudia Lötbecke, Köln, und Prof. Dr. Christoph Zacharias, Köln

PD Dr.-Ing. habil. Martin Engelien  
Dipl.-Inf. Jens Homann (Hrsg.)

# Virtuelle Organisation und Neue Medien 2002

Workshop GeNeMe2002  
Gemeinschaften in Neuen Medien

TU Dresden, 26. und 27. September 2002



**JOSEF EUL VERLAG**  
Lohmar · Köln

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Virtuelle Organisation und Neue Medien 2002 / Workshop GeNeMe 2002 – Gemeinschaften in Neuen Medien – TU Dresden, 26. und 27. September 2002. Hrsg.: Martin Engeliens ; Jens Homann. – Lohmar ; Köln : Eul, 2002

(Reihe: Telekommunikation und Medienwirtschaft ; Bd. 14)

ISBN 3-89936-007-9

© 2002

Josef Eul Verlag GmbH

Brandsberg 6

53797 Lohmar

Tel.: 0 22 05 / 90 10 6-6

Fax: 0 22 05 / 90 10 6-88

<http://www.eul-verlag.de>

[info@eul-verlag.de](mailto:info@eul-verlag.de)

Alle Rechte vorbehalten

Printed in Germany

Druck: RSP Köln

**Bei der Herstellung unserer Bücher möchten wir die Umwelt schonen. Dieses Buch ist daher auf säurefreiem, 100% chlorfrei gebleichtem, alterungsbeständigem Papier nach DIN 6738 gedruckt.**



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

PD Dr.–Ing. habil. Martin Engelen

Dipl.–Inf. Jens Homann

(Hrsg.)

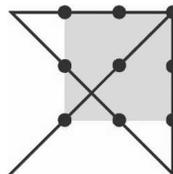


an der

Fakultät Informatik der Technischen Universität Dresden

in Zusammenarbeit mit der  
Gesellschaft für Informatik e.V.,  
GI-Regionalgruppe Dresden

gefördert von der Klaus Tschira Stiftung  
gemeinnützige Gesellschaft mit beschränkter Haftung



am 26. und 27. September 2002

in Dresden

<http://pdai.inf.tu-dresden.de/geneme>

Kontakt: Thomas Müller ([geneme@pdai.inf.tu-dresden.de](mailto:geneme@pdai.inf.tu-dresden.de))

---

## **D.4. KaViDo - Ein web-basiertes System für kooperative Forschungs- und Entwicklungsprozesse**

*Oliver Taminé*

*Prof. Dr. Rüdiger Dillmann*

*Institut für industrielle Anwendungen der Informatik und Mikrosystemtechnik*

*Universität Karlsruhe*

### **Kurzzusammenfassung**

*Im folgenden wird ein web-basiertes System namens KaViDo für die kollaborative Forschung und Entwicklung vorgestellt. Nach der Erläuterung, weshalb Bedarf für ein solches System besteht, wird dessen Aufbau geschildert. Im Detail werden die drei Schichten (Präsentationsschicht, Entwicklungsschicht und Persistenzschicht) vorgestellt. Die Zielsetzungen des KaViDo-Systems sind die Aufzeichnung der Entwicklungsprozesse, das Verwalten der Kompetenzen der verteilten Experten, der Austausch von Entwicklungserfahrungen sowie eine Assistenz bei der Produktentwicklung.*

*Der praktische Einsatz von KaViDo wird durch ein interdisziplinäres Projektpraktikum demonstriert. Verschiedene Studentengruppen hatten die Aufgabe, mittels KaViDo einen Roboter zu entwerfen, konstruieren und programmieren.*

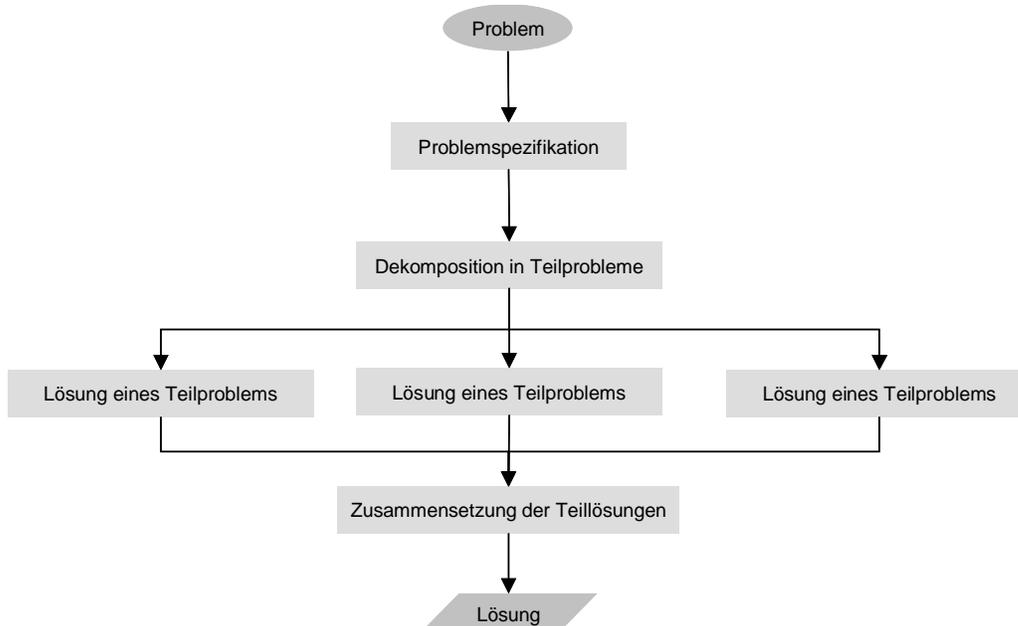
### **1. Motivation und Zielsetzung**

Die Globalisierung der Märkte und die immer schlechtere Auftragslage in der Wirtschaft führen zu einem ständig wachsenden Wettbewerbsdruck. Durch die Entwicklung leistungsfähiger Produkte versuchen die Unternehmen, ihre Überlebensfähigkeit zu sichern. Um die höheren Anforderungen an den Produktentwicklungsprozess zu erfüllen, ist es häufig erforderlich, Experten aus unterschiedlichen natur- und ingenieurwissenschaftlichen Disziplinen einzubinden. Hierdurch ergeben sich eine Reihe neuer Anforderungen an die Entwicklungsprozesse, die bislang noch nicht ausreichend untersucht sind.

Zielsetzung dieser Untersuchungen sind eine Unterstützung verteilter Expertenteams und die Weitergabe von Erfahrungswissen. Das System KaViDo führt die Aufzeichnung und Dokumentation von Entwicklungsprozessen durch. Hierdurch ist es möglich, Muster von erfolgreichen Prozessen zu entwerfen. Das System bietet ebenfalls Möglichkeiten zur Suche nach erforderlichen Kompetenzen und ermöglicht den Austausch von Erfahrungserfahrungen [TaDi99a].

## 2. Modellierung der Entwicklungsprozesse

Der Beginn eines Entwicklungsprozesses ist grundsätzlich die Problemstellung, die es zu lösen gilt. In Anlehnung an [Ade96] kann ein verteilter Entwicklungsprozess in vier Teilschritte untergliedert werden.



**Abb. 1: Struktur eines verteilten Entwicklungsprozesses**

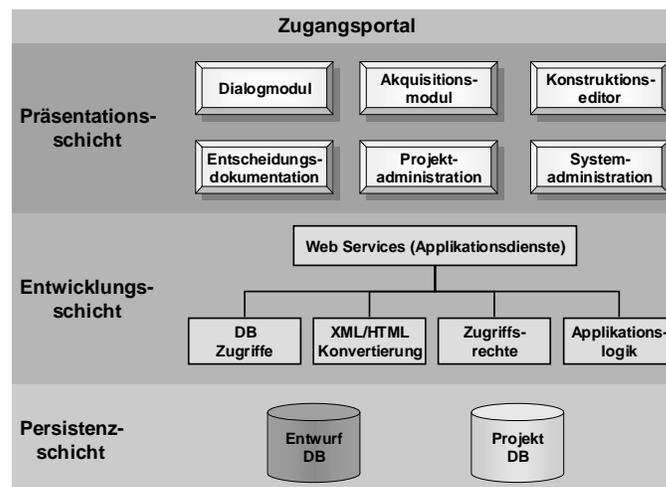
Schritt eins ist die Problemspezifizierung, bei der mehrere Experten das Problem definieren und mögliche Lösungsvorschläge diskutieren. Als nächstes wird das Problem in einzelne Teilprobleme zerlegt. Darauf folgt deren Abarbeitung und letztlich die Zusammensetzung der Teillösungen. Sie stellt das Ergebnis dar (s. Abb. 1).

KaViDo unterstützt jeden dieser vier Schritte. Dabei geht das System von der Annahme aus, dass das Ziel eines Entwicklungsprozesses entweder der Entwurf einer Komponente oder die Kombination von Komponenten ist [TaDi99a]. KaViDo unterteilt die Bearbeitung eines solchen Entwicklungsprozesses in drei Schritte. Erster Schritt ist die Schemasuche. Grundlage dieser Suche sind die Anforderungen an das zu entwickelnde Produkt. Ziel der Suche ist es, eventuell früher schon einmal praktizierte und erfolgreiche Lösungen wiederzufinden. Dabei beschreibt ein Schema, welche Komponenten zur Lösung eines spezifischen Problems erforderlich sind. Existiert kein solches Schema oder muss dieses angepasst werden, erfolgt eine eigene Komponentenauswahl. Dies stellt Schritt zwei eines Entwicklungsprozesses dar. Eine Komponente stellt eine abstrakte Beschreibung dar, die die erforderlichen Eigenschaften definiert, welche eine gegebene Zielsetzung innerhalb eines gegebenen Kontextes

erfordert. Ein Beispiel einer abstrakten Komponente stellt eine Batterie dar. Ihre Eigenschaften sind die Spannung und Stromstärke, aber auch Betriebsdauer, Größe Form, usw. Im letzten Schritt erfolgt die Ersetzung der abstrakten Komponenten durch reale Produkte. Hierzu wird ein Produktkatalog bereitgestellt, der anhand der geforderten Eigenschaften eine Vorauswahl liefert.

### 3. Systemaufbau

Neben den zuvor beschriebenen Funktionalitäten ist beim Systemaufbau noch eine nicht-funktionale Anforderung wesentlich. Das System darf auf Seite des Clients keinerlei Programminstallationen benötigen<sup>1</sup>. Somit wird zur Anwendung lediglich ein herkömmlicher WWW-Browser vorausgesetzt und mittels des Protokolls http zwischen Client und Server kommuniziert. Dies gewährleistet einen einfachen und schnellen Zugriff aller Projektbeteiligter [TaDi00].



**Abb. 2: Die Komponenten des KaViDo-Systems**

KaViDo stellt ein Zugangsportaal dar, welches die oben erläuterten Schritte unterstützt und dabei die Kooperation verteilter Experten über das Internet ermöglicht. Beispielsweise können mehrere Personen an der Anforderungsspezifizierung bzw. der Komponentenauswahl gemeinsam arbeiten. KaViDo ist in drei Schichten unterteilt (s. Abb. 22). Die Präsentationsschicht umfasst sechs verschiedene Client-Programmmodule. Sie dienen dem Anlegen neuer Muster, der Abfrage von Komponenten und Produkten

<sup>1</sup> Die Programmmodule des KaViDo-Systems sind in Java geschrieben. Hieraus ergibt sich, dass auf Client-Seite eine Java Virtual Machine benötigt wird.

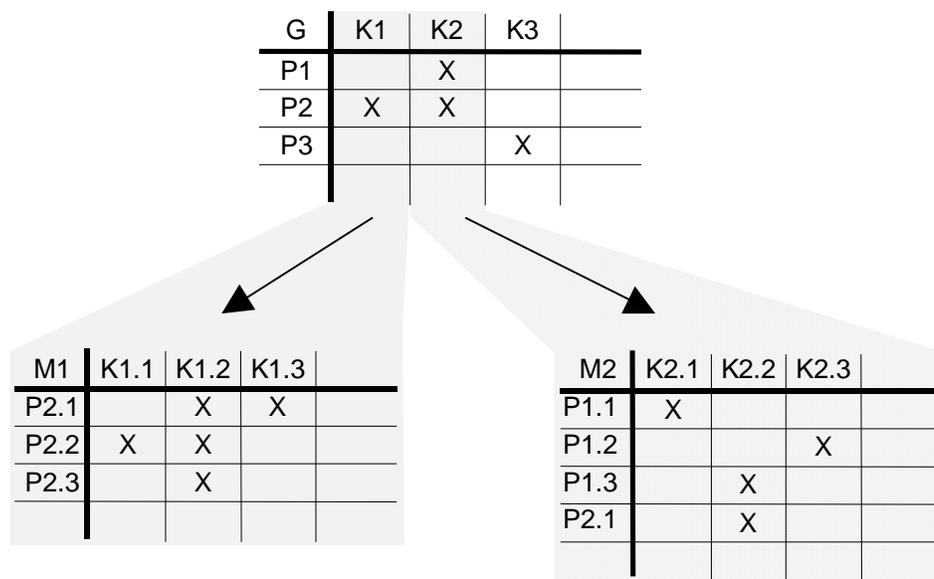
sowie einem Modul zur graphischen Konstruktionsunterstützung. Die Entwicklungsschicht beinhaltet alle serverseitigen Applikationsdienste. Unter ihr befinden sich die Datenbanken der Persistenzschicht.

### **3.1 Persistenzschicht**

Die Persistenzschicht besteht aus zwei Datenbanken. Dies ist zum einen eine Projektdatenbank, in der alle projektspezifischen Daten abgelegt werden, und zum anderen eine Entwurfsdatenbank, welche alle entwurfsrelevanten Daten enthält, deren Lebensdauer über die Projektabwicklung hinweg relevant ist.

Die Projektdatenbank speichert sämtliche personenbezogenen Informationen, die Kompetenzen der Entwickler und deren Rolle. Eine Rolle legt die Verantwortlichkeiten einer Person fest. Zur Bestimmung dieser Verantwortlichkeiten wird ein Produkt anhand seiner Funktionen strukturiert. Jede Produktfunktion beschreibt eine Produkthanforderung. Dabei kann jede dieser Produktfunktion iterativ in weitere Teilproduktfunktionen zerlegt werden. Dies führt zu einem Baum von Produktfunktionen, welcher sämtliche Produkthanforderungen enthält [Info99].

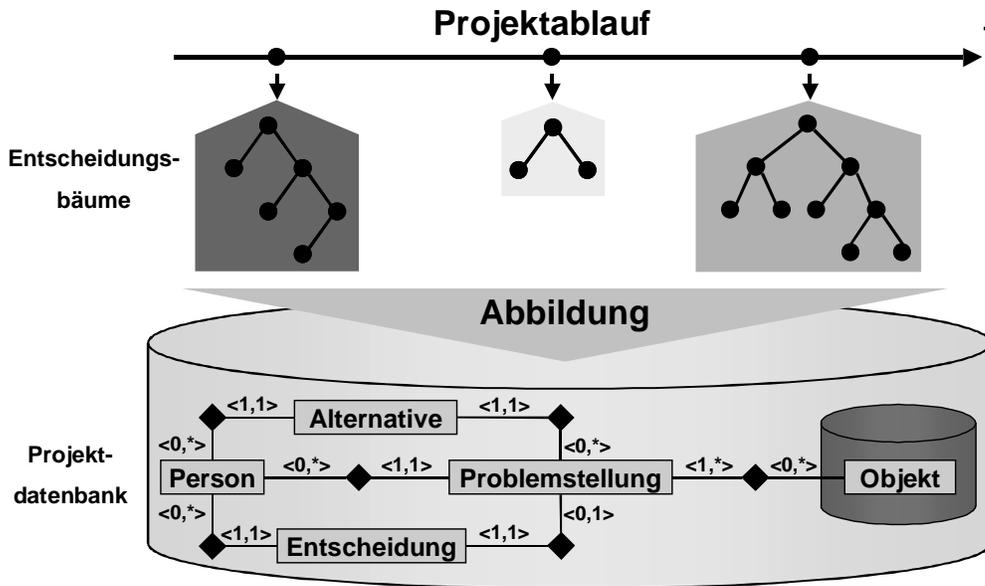
In einer Informationsmatrix wird eine Zuordnung von Produktfunktionen zu Kompetenzzentren festgehalten. Die Spalten einer Informationsmatrix sind mit den Bezeichnungen der Kompetenzzentren überschrieben. Die einzelnen Zeilen der Matrix werden durch die Funktionen des zu entwickelnden Produktes beschrieben. Für jedes Kompetenzzentrum sind in dessen Spalte die Produktfunktionen markiert, die für seinen Kooperationsbeitrag wichtig sind.



**Abb. 3: Informationsmatrizenbaum**

Die Informationsmatrizen mit ihren Produktfunktionen und Kompetenzzentren werden in einem Baum gemäß ihrer Zerlegung angeordnet. Die Knoten des Matrixbaumes bestehen aus den Informationsmatrizen. Die Wurzel des Matrixbaumes heißt Grundmatrix (G). Diese enthält in ihren Spaltenüberschriften die Namen der Kompetenzzentren (K) und in ihren Zeilenüberschriften die Produktfunktionen (P) (s. Abb. 33).

Die Projektentscheidungen werden mit Hilfe von Entscheidungsbäumen dokumentiert [BSH99]. Dabei umfasst ein solcher Entscheidungsbaum das Problem, seine Lösungsalternativen und eine Bewertung jeder Alternative [TaDi99b].



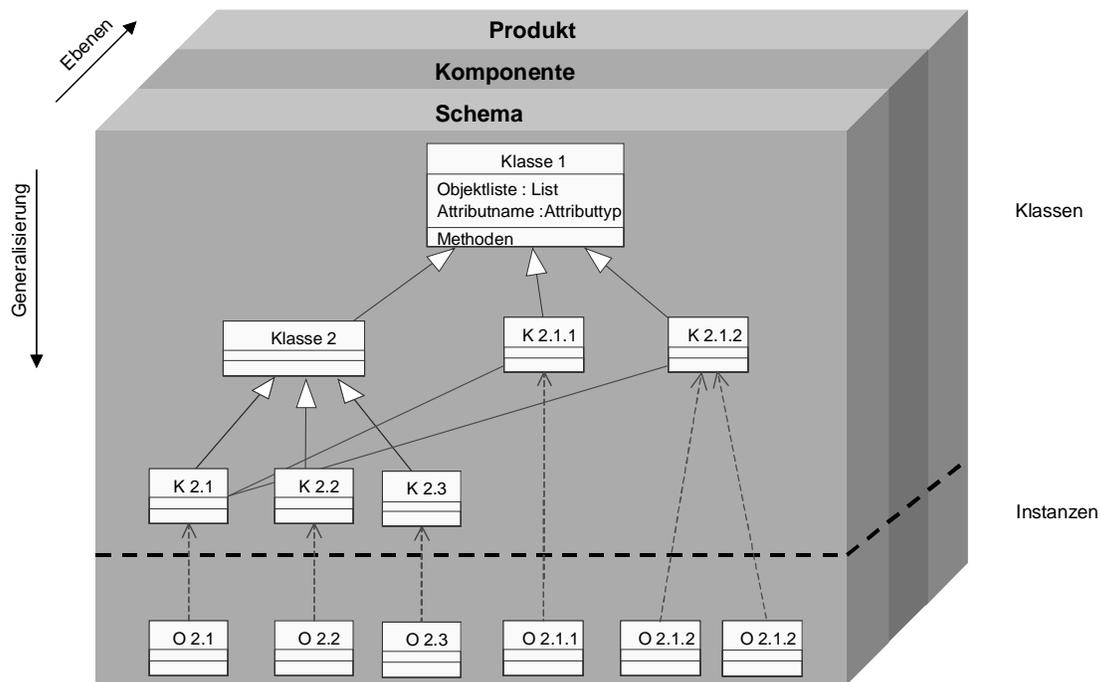
**Abb. 4: Datenbankmodellierung der Entscheidungsbäume**

Sämtliche projektbezogenen Daten werden in einer relationalen Datenbank abgelegt. Abb. 44 illustriert die Ablage von Entscheidungsbäumen in der Projektdatenbank. Hierzu ist ein kleiner Auszug der Projektdatenbank mittels eines ER-Diagramms beschrieben. Optional besteht die Möglichkeit, einen Entscheidungsbaum mit einem Objekt der Entwurfsdatenbank zu verbinden. Dies ist sinnvoll, falls sich eine Problemstellung einer spezifischen Komponente zuordnen lässt.

Zur Speicherung der Entwurfsdaten wird eine objekt-orientierte Datenbank verwendet. Sie beinhaltet die Daten, die sich nicht direkt zum Projekt zuordnen lassen. Im einzelnen sind dies Schemata, Komponenten und Produkte (s. Abschnitt 2).

Die Repräsentation der Entwurfsdatenbank besteht aus den Ebenen Schema, Komponente und Produkt (s. Abb. 5).

Jede Ebene besitzt ihr eigenes Objektmodell. Untereinander sind diese Ebenen mittels Assoziationen auf Objektinstanzenebene verbunden. In Anlehnung an [EsJe90] sind zwischen den Objekten der gleichen Ebene zwei Beziehungsarten definiert. Die Generalisierung wird verwendet um die Struktur von Komponenten und Unterkomponenten abzubilden. Eine Assoziation auf der gleichen Ebene bedeutet, dass ein Objekt ein anderes Objekt der gleichen Ebene unbedingt erfordert.



**Abb. 5: Entwurfsdatenbank**

Jede Ebene besteht aus einem dynamischen Objektmodell. Hierdurch ist es während des Programmablaufs möglich, Klassen und Objektmethoden zu entwickeln und zu kompilieren. Ebenso ist das Hinzufügen, Löschen und Aktualisieren von Klassen und Instanzen während der Laufzeit möglich.

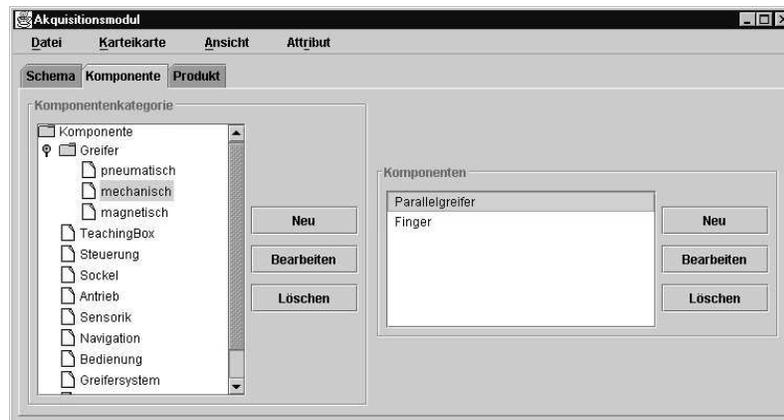
### 3.2 Entwicklungsschicht

Die Entwicklungsschicht ist eine abstrakte Schicht ohne direkte Benutzerinteraktion. Sie ist verantwortlich für den Dialog zwischen den Anwendungsmodulen auf der Client-Seite und den Datenbanken auf Server-Seite. Hierzu sind in dieser Schicht eine Reihe von Webdiensten implementiert. Die Kommunikation zwischen Client und Server wird mittels http als Kommunikationsprotokoll abgewickelt. Zum Aufrufen der Webdienste wird SOAP eingesetzt [TaAm01].

Die Entwicklungsschicht umfasst vier Teile. Teil eins besteht aus den Webdiensten zum Zugriff auf die relationale und objekt-orientierte Datenbank der Persistenzschicht. Teil zwei ist verantwortlich für die Datenausgabe. Sie konvertiert die Daten der Persistenzschicht in XML oder HTML mittels XSLT. Benutzer- und Zugriffsrechte werden mit Teil drei der Webdienste verwaltet. Die letzte Kategorie der Webdienste umfassen alle anwendungsmodulspezifischen Funktionen.

### 3.3 Präsentationsschicht

Die Präsentationsschicht besteht aus insgesamt sechs Anwendungsmodulen. Das Modul Projektadministration verwaltet alle projektspezifischen Verwaltungsdaten, z.B. Personen, Rollen, Kompetenzen, Produktfunktionen und Informationsmatrizen (s. Abschnitt 3.1). Die Systemadministration regelt die Sicherung der Datenbanken, kontrolliert und verwaltet den Festplattenspeicher, usw.



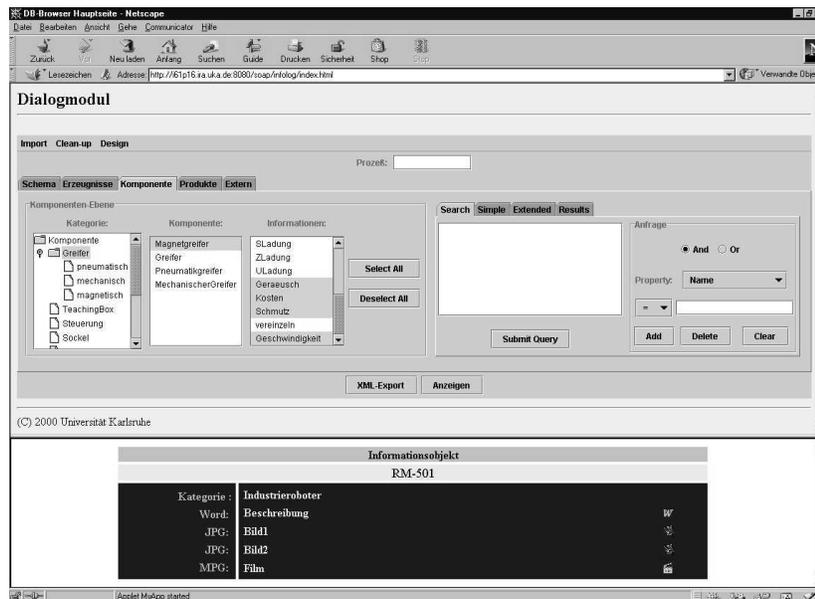
**Abb. 6: Akquisitionsmodul**

Das Akquisitionsmodul ermöglicht das Einfügen, Aktualisieren und Löschen von Entwurfsdaten. Es visualisiert die drei Ebenen der Entwurfsdatenbank (s. Karteikarten in Abb. 6). Die linke Fensterhälfte zeigt die Klassen der selektierten Ebene, während in der rechten Fensterhälfte die korrespondierenden Objektinstanzen aufgelistet werden.

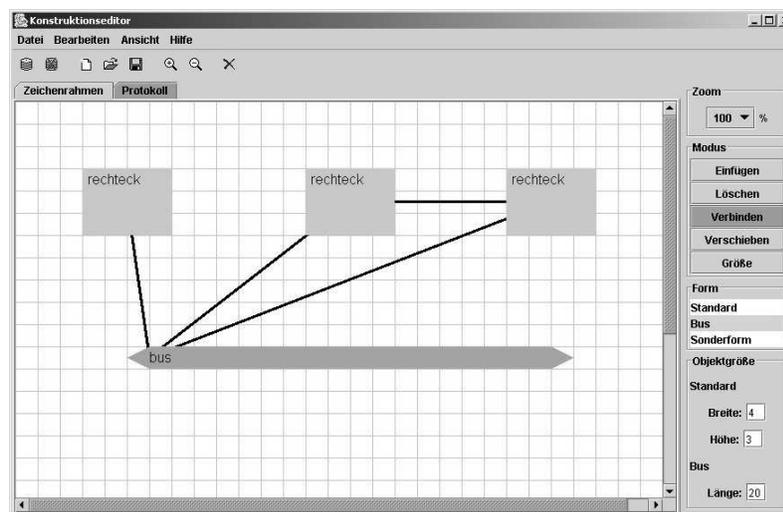
Das Dialogmodul stellt den Navigator für die Entwurfsdatenbank dar. Seine Aufgaben sind die Aufbereitung der Daten in eine benutzerfreundliche Ausgabe, die einfache und schnelle Navigation innerhalb der Datenbank und die Suche nach Datenobjekten.

Abb. 7 zeigt den Bildschirmaufbau des Dialogmoduls. Das Fenster ist in zwei Bereiche unterteilt. Die obere Fensterhälfte ist das Steuerungsfenster, während im unteren Bereich die Entwurfsdaten visualisiert werden. Das Steuerungsfenster besitzt fünf Karteikarten. Die Karteikarten Schema, Komponente und Produkte nehmen Bezug zu den entsprechenden Ebenen der Entwurfsdatenbank. Die Karteikarte Erzeugnisse listet alle realisierten Produkte auf und bietet deren Beschreibung an. Mittels der Karteikarte Externe Daten können externe Datenquellen eingebunden werden. Jede Karteikarte ist in zwei Teilbereiche untergliedert. Die rechte Seite umfasst die Suchmaske, welche drei verschiedene Suchmethoden anbietet. Die linke Seite enthält den Datenbanknavigator.

Er unterteilt die Daten in Klassen und Objekte. Die Visualisierungsfensterhälfte zeigt die Daten in HTML an. Zuvor wurden diese von XML in HTML mittels der Entwurfsschicht transformiert (s. Abschnitt 3.2).



**Abb. 7: Dialogmodul**



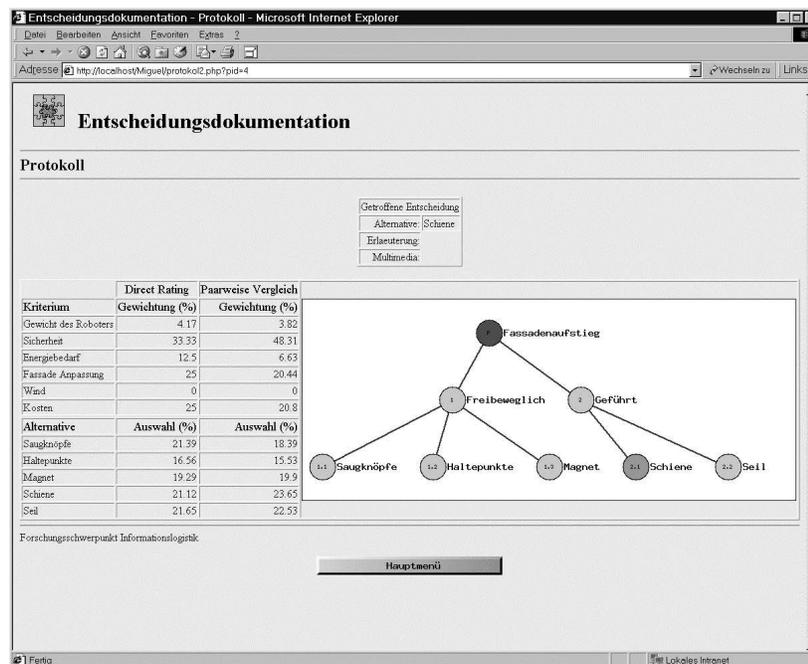
**Abb. 8: Konstruktionseditor**

Der Konstruktionseditor ist ein Werkzeug zur Unterstützung des Entwurfsprozesses und zur dessen Dokumentation (s. Abb. 8). Hierzu benutzt er die Entwurfsdatenbank zur Selektion von Schemata, Komponenten und Produkten. Gleichzeitig wird dabei ein Entwurfsprotokoll erstellt und in der Projektdatenbank abgelegt. Dieses Protokoll um-

fasst eine Beschreibung jeder ausgeführten Aktion inklusive Datum, Uhrzeit und der Person, die die Aktion ausgeführt hat.

Der Editor beschreibt eine Konstruktion anhand eines Blockdiagramms. Ein Schema stellt ein solches Diagramm dar, welches die notwendigen Komponenten beinhaltet. Es kann um weitere Komponenten erweitert oder mit anderen Schemata kombiniert werden. Die darin enthaltenen Komponenten müssen weiter detailliert und im letzten Schritt durch Produkte ersetzt werden. Hierzu offeriert die Entwurfsdatenbank eine Liste von möglichen und nützlichen Teilkomponenten und Produkten. Sollten während dieses Ersetzungsschrittes Probleme auftreten, bietet das Werkzeug zur Entscheidungs-dokumentation weitere Unterstützung an.

Abb. 9 zeigt einen bereits fertig entwickelten Entscheidungsbaum eines Konstruktionsproblems. Für die Erstellung eines solchen Entscheidungsbaums ist eine detaillierte Problemspezifikation notwendig, welche alle problemrelevanten Sachgebiete definiert. Die Entscheidungsdokumentation informiert anhand der Sachgebiete alle betreffenden Entwicklungsexperten automatisch per Mail.



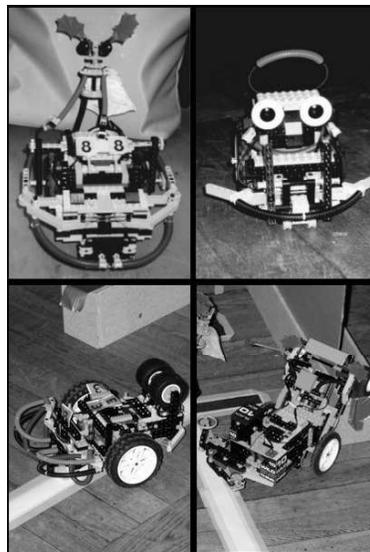
**Abb. 9: Entscheidungsdokumentation**

Diese Experten entwickeln über das World Wide Web die Äste des Entscheidungsbaums. Jeder Ast stellt dabei eine mögliche Lösungsalternative dar. Sind genügend Lösungsalternativen entwickelt oder ist der Zeitpunkt einer Problemlösung

fällig, erfolgt die Alternativenauswahl. Sie kann auf unterschiedliche Weisen durchgeführt werden. Es gibt verschiedene Abstimmungsverfahren (Direkte Bewertung, Paarvergleich, analytischer hierarchischer Prozess). Es kann auch ein Entscheider festgelegt werden. Die gewählte Alternative wird innerhalb eines Entscheidungsbaums grün markiert. Hierdurch dokumentiert das Modul umfassend das Problem, die verschiedenen Lösungsalternativen des Problems und die Argumente, welche für bzw. gegen eine Alternative sprechen.

#### 4. Einsatzgebiete

KaViDo kann sowohl in der Forschung als auch in der Lehre eingesetzt werden. Bislang wird es im Rahmen eines interdisziplinären Praktikums an der Universität Karlsruhe eingesetzt [TaDi01]. Teilnehmer sind ca. 50 Studenten der Fakultäten Maschinenbau, Architektur und Informatik. Sie werden in zehn Gruppen aufgeteilt und müssen gemeinsam innerhalb von zehn Tagen einen Roboter bauen, der vorgegebene Aufgaben erfüllen muss (s. Abb. 10).



**Abb. 10: Mittels KaViDo konstruierte Roboter**

Jedes Studententeam umfasst drei Fachgebiete. Zur Erfüllung der Aufgabe besteht für jede Gruppe die Notwendigkeit, ihr unterschiedliches Fachwissen zu kombinieren. Dabei gilt es, die Schwierigkeiten beim Austausch von Wissen unterschiedlicher Spezialgebiete zu überwinden. Somit erfordert das Praktikum neben dem Arbeiten in selbstorganisierenden Teams auch das kooperative Lernen.

Die Dauer des Praktikums beträgt zehn Tage, weshalb für die Umsetzung der Aufgabe nur wenig Zeit bleibt. Da das Praktikum eine Reihe von Schnittstellen zwischen den Studentengruppen und ihren unterschiedlichen Kompetenzen beinhaltet, ist es gut für den Einsatz von KaViDo geeignet.

Mittels KaViDo können die Studenten ihre Konstruktionen dokumentieren, verschiedene Problemstellungen und Lösungsvorschläge diskutieren sowie Schemata bereits entworfener Roboter verwenden. Durch die Bereitstellung früherer Erfahrungen gelingt den Studenten trotz des Zeitdrucks eine stetige Verbesserung und Leistungssteigerung der entworfenen Roboter.

## 5. Literatur

- [Ande96] Anderl, R.: *Collaborative Work in Concurrent Design*; ESPRIT Technical Day on Advanced and Concurrent Engineering, European Commission, Brüssel, 1996.
- [BSH99] Bhargava, H.; Sridhar, S.; Herrick, C.: *Beyond Spreadsheets: Tools for Building Decision Support Systems*; Computer – Innovative Technology for Computer Professionals; IEEE; März 1999, S. 31-39.
- [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.
- [Info99] Blodau, G.; v. Both, P.; Klimesch, C.; Lanza, M.; Ostermayer, R.; Rude, S.; Taminé, O.; Ziegler, P.: *Informationslogistik – Rechnerunterstützte branchenübergreifende Kooperation*; Herausgeber: Grabowski, H.; Rude, S.; B. G. Teubner-Verlag, Stuttgart; 1999.
- [TaAm01] Taminé, O., Armbruster, M.: *Web-Service Entwicklung mit SOAP, XML in Action*, München, 2001.
- [TaDi99a] 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.
- [TaDi99b] Taminé, O.; Dillmann, R.: *Multimediale Protokollierung interdisziplinärer und gruppenbasierter Entscheidungsprozesse*; in: VDI-Fortschritt-Berichte Nr. 156, 11. Forum Bauinformatik, Darmstadt, 1999.

- 
- [TaDi00] Taminé, O.; Dillmann, R.: *Eine Kommunikationsarchitektur für den Wissensaustausch in interdisziplinären Projekten*; In: VDI Fortschritt-Berichte, Forum Bauinformatik 2000; VDI Verlag, Berlin; 2000.
- [TaDi01] Taminé, O.; Dillmann, R.: *An interdisciplinary student contest with LEGO robots*; In Proceedings of 1st Workshop on Robotics Education and Training; Weingarten; 2001.

