Modellierung von Hypermedia-Applikationen mit *HyDev*

Peter Pauen, Josef Voss, Hans-Werner Six

Praktische Informatik III, FernUniversität Hagen, D-58084 Hagen
email: {peter.pauen | josef.voss | hw.six@fernuni-hagen.de}

Positionspapier, Januar 1998

Zusammenfassung. Die vorliegende Arbeit stellt den *HyDev*-Ansatz zur strukturierten Entwicklung von typischen Hypermedia-Applikationen vor. *HyDev* konzentriert sich auf die frühen Phasen des Entwicklungsprozesses, deren Ergebnisse und Entscheidungen in Form dreier Modelle festgehalten werden. Der Schwerpunkt dieses Papiers liegt auf *HyDev*s Domainmodell.

Stichworte. Hypermedia, Multimedia, Authoring, Software Engineering, Requirements Engineering, Modellierung, Modellbasierte Software-Entwicklung

1 Einleitung

In den letzten Jahren nahmen Bedeutung und Verbreitungsgrad von Hypermedia-Applikationen - im folgenden kurz HMA genannt - deutlich zu. Es existieren zahlreiche verschiedene Arten von HMA, z.B. elektronische Bücher, multimediale Lern-/Lehrsoftware sowie Produktkataloge und -präsentationen. Diese besitzen eine Reihe charakteristischer Gemeinsamkeiten, die sie von konventioneller Software unterscheiden.

Ein wichtiges Kennzeichen ist, daß die Objekte und Informationen des Programms multimedial in Form von Texten, Graphiken, Bildern, Videos, Audios, Animationen präsentiert werden. Mithin kann die Präsentation von Objekten eine zeitliche Ausdehnung besitzen. Im Unterschied zu herkömmlichen Programmen mit direkt-manipulativen Oberflächen können HMA hinsichtlich der zu präsentierenden Inhalte eine bestimmte Erzählstruktur sowie Regie aufweisen. Die wichtigsten Benutzeroperationen sind komfortable Navigationen zur Auswahl der zu präsentierenden Objekte. Typische Informationsverarbeitung ist dagegen von sekundärer Bedeutung, denn das Erzeugen neuer Objekte, die Manipulation vorhandener Objekte sowie Berechnungen spielen einer eher untergeordnete Rolle.

In der Regel besitzen auch HMA eine ganz erhebliche Komplexität. So weisen bspw. gängige Computerspiele gemeinhin eine komplizierte endogene Struktur, eine anspruchsvolle audiovisuelle Gestaltung und einen hohen Interaktionsgrad auf. Daraus leitet sich die Forderung nach einem systematischen Vorgehen bei der Entwicklung ab, welches den Spezifika dieser Art von Software Rechnung trägt. Ohne ein strukturiertes, d.h. planvolles und systematisches Vorgehen läuft die Entwicklung von HMA schnell Gefahr, im Produktions-Chaos zu enden. Die Anwendung von Software-Engineering-Methoden ist also auch in diesem Bereich grundsätzlich sinnvoll und angebracht.

Gleichwohl entstehen die meisten HMA auch heutzutage noch quick&dirty ohne einen besonderen methodischen Rahmen mit den entsprechenden fatalen Folgen für Zuverlässigkeit und Wartbarkeit des Endproduktes. Zwar kennt das klassische Software Engineering mittlerweile zahlreiche erprobte Methoden und Ansätze, jedoch lassen sich diese aufgrund der andersgearteten Natur dieser Art von Software nur bedingt für deren Entwicklung einsetzen. Aber auch die vereinzelt vorgeschlagenen spezielleren Ansätze [1] erweisen sich letztlich als zu schwach, da zu technisch. Meist handelt es sich dabei um den Versuch, eine HMA ohne vorherige Analyse in einem einzigen Schritt zu spezifizieren. Da bei der Spezifikation einer HMA jedoch zahlreiche Aspekte zu berücksichtigen sind, besteht hier schnell die Gefahr einer Überfrachtung mit zu vielen Details. Und die zahlreichen kommerziellen Autorensysteme schließlich konzentrieren sich allein auf die eigentliche Realisierung und unterstützen die vorbereitenden Analyse- und Entwurfstätigkeiten in keiner Weise. Folgerichtig beklagen sich Praktiker wie bspw. Multimedia-Producerin Kathy Kozel [2] über das Fehlen jeglicher ingenieursmäßiger Vorgehensweisen und die daraus resultierenden negativen Konsequenzen für die praktische Entwicklungsarbeit.

Vor dem Hintergrund dieser Beobachtungen haben wir den *HyDev*-Ansatz[1] zur strukturierten, d.h. systematischen Entwicklung typischer HMA entwickelt. Die wesentlichen Merkmale von *HyDev* stellt Kapitel 2 vor. Das abschließende dritte Kapitel spricht derzeit noch offene Fragen an und gibt Einblick in den aktuellen Stand sowie unsere zukünftigen Arbeiten.

2 Der *HyDev*-Ansatz

2.1 Überblick

Anforderungen. HMA entstehen aufgrund ihrer i.d.R. hohen strukturellen Komplexität typischerweise in einem langwierigen Entwicklungsprozeß. Jede strukturierte Entwicklung von HMA muß daher eine explizite Requirements Engineering Phase aufweisen, in der die wesentlichen Aspekte des zu entwickelnden Systems ermittelt und in nicht allzu technischer Form übersichtlich dokumentiert werden. Die Beherrschung der Komplexität gelingt dabei allein durch Abstraktion, d.h. Beschränkung auf ausgewählte wichtige Aspekte. Eine hohe Bedeutung besitzt zudem die Implementierungsunabhängigkeit der Ergebnisse dieser Phase, d.h. es sollte keine Herangehensweise für die eigentliche Realisierung präjudiziert werden. Somit sollten bspw. weder Annahmen hinsichtlich der Medienobjekte noch Vorgaben bzgl. der Frage gemacht werden, ob die HMA mit Hilfe eines (vielleicht gar bestimmten) Autorensystems realisiert oder konventionell mittels einer speziellen Programm-Bibliothek programmiert werden soll. Ebenfalls wichtig ist, daß die rasche Entwicklung von Prototypen unterstützt wird.

Beobachtungen und Schlußfolgerungen daraus. Charakteristisch für HMA ist, daß sich ihre grundlegende Struktur aus ihren Instanzen und den zugehörigen Beziehungen ableitet. Man denke bspw. an eine multimediale Präsentation wie sie auf der Rolling Stones CD Stripped abgelegt ist. Dort stehen Einzelobjekte wie ein Plattenkatalog, die Band an sich, ihre vier Mitglieder, bestimmte Interviews und Songs sowie zwei Musikvideos im Vordergrund. Im Gegensatz zu konventioneller Software reicht hier ein ER-Modell oder ein Klassen-Modell allein nicht aus. Vielmehr müssen die der Anwendung zugrundeliegenden <u>Objekte</u> in einem Instanzenmodell explizit erfaßt werden.
Da die Betrachtung von Objekten aber nur dann förderlich und sinnvoll ist, wenn zuvor Klassen sauber modelliert wurden, ist auch ein Klassenmodell erforderlich. Dieses muß jedoch <u>angepaßt</u> sein, denn in HMA treten besondere Arten von Objekten auf[2]. Insgesamt fehlt aber eine geeignete Domainmodellierung, z.B. in Form einer angepaßten oder erweiterten Variante von OOA-Methoden, bisher ersatzlos.
An ein Domain- und Instanzenmodell kann sich allerdings die eigentliche Realisierung mit konkreten Medienobjekten nicht direkt anschließen. Hier bleibt ein erheblicher Gestaltungs- und Entscheidungsspielraum, der in wesentlichen Punkten verbindlich eingegrenzt werden sollte. Derartige Festlegungen müssen allerdings von Details wie konkreten Layout- und Zeitangaben oder Farben abstrahieren. Denn zum einen würde eine zu frühe Detaillierung bereits die eigentliche Realisierung präjudizieren oder gar vorwegnehmen und einen unnötig hohen Revidierungsaufwand nach sich ziehen. Zum anderen dient eine solche Abstraktion zweifellos der Beherrschung der hohen Komplexität. Vereinfacht ausgedrückt muß daher für die Objekte des Programms angegeben werden, in welcher Form sie dem Benutzer in der laufenden Applikation präsentiert werden, d.h. ob als Text, Graphik, Video, VRML-Welt o.ä. Weitere Festlegungen betreffen die Navigationsstruktur sowie Benutzerinteraktionen. Für derartige <u>Aspekte der Objektrepräsentation</u> wird ein eigenständiges, spezielles Repräsentationsmodell benötigt. Für ein solches Modell spricht zudem, daß sich Beziehungen zwischen Objekten eben nicht grundsätzlich 1:1 auf Strukturen zwischen Objektrepräsentationen abbilden lassen, da ein Objekt durchaus mehrere unterschiedliche Repräsentationen besitzen kann. Umgekehrt ist es möglich, daß mehrere Objekte eine einzige gemeinsame Repräsentation besitzen. Andere Ansätze wie RMM und HDM [3,4,5] verfolgen diese Trennung zwischen Objekten und ihren Repräsentationen nicht, obwohl inhaltliche Struktur und Navigationsstruktur große Unterschiede aufweisen können.

Konsequenzen für *HyDev*. Entsprechend den vorgenannten Beobachtungen ist *HyDev* ein modellbasierter Ansatz, der drei spezialisierte, aufeinander aufbauende Modelle kennt: ein *Domainmodell* für die Domain-Analyse, ein *Instanzenmodell* zur Erfassung der zugrundeliegenden Objekte sowie ein *Repräsentationsmodell* zur Beschreibung der Objektrepräsentationen und Navigationen zwischen diesen. Jedes dieser Modelle betrachtet die zu entwickelnde Applikation aus einem anderen Blickwinkel, was nicht zuletzt der Beherrschung der Komplexität dient. In den folgenden Unterkapiteln werden die drei *HyDev*-Modelle detaillierter vorgestellt, wobei der Schwerpunkt auf dem Domainmodell liegt.

1. In das Akronym *HyDev* sind die Worte <u>Hy</u>permedia und <u>Dev</u>elopment eingeflossen.
2. In Unterkapitel 2.2 wird darauf ausführlich eingegangen.

2.2 Domainmodell

HyDev basiert auf der Vorstellung, daß einer HMA eine Sammlung diverser Objekte zugrundeliegt, die in u.U. komplexer Weise miteinander in Beziehung stehen. Im Rahmen der Domain-Analyse werden die Klassen dieser Objekte sowie die zugehörigen Beziehungen in einem Domainmodell erfaßt. Aufgrund der besonderen Natur von HMA lassen sich verschiedene Arten von Klassen unterscheiden. Im einzelnen kennt das *HyDev*-Domainmodell daher:

- *Klassen für narrativ strukturierende Einheiten (N-Klassen)*
- *Klassen für Objekte mit räumlicher Ausdehnung (S-Klassen)*
- *Klassen für Agenten (A-Klassen)*
- *Konventionelle Klassen für alle sonstigen Objekte*

Zwischen diesen Klassen können neben den üblichen Vererbungsbeziehungen, Assoziationen und Ganzes-Teile-Beziehungen auch zahlreiche spezialisierte Beziehungen bestehen, die eine adäquatere Domainmodellierung als bspw. konventionelle OOA- oder ER-Modellierung erlauben.

Klassen für narrativ strukturierende Einheiten (N-Klassen). HMA können hinsichtlich ihrer thematischen Inhalte strukturiert sein, d.h. eine *Erzählstruktur* aufweisen. Einer solchen Erzählstruktur liegen spezielle Objekte zugrunde, die eine Art Drehbuchcharakter haben. Mit Hilfe dieser Objekte - *Erzähleinheiten* genannt - werden typischerweise Abläufe modelliert. Erzähleinheiten weisen damit zum einen selbst eine Erzählstruktur auf und können zum anderen über Beziehungen zu anderen Erzähleinheiten komplexe Erzählstrukturen bilden. Zu ihrer Erfassung dienen die N-Klassen des Domainmodells. Zwischen diesen bzw. ihren Objekten können die folgenden spezialisierten Beziehungen bestehen:

- *Folge-Beziehungen, mit denen die inhalt-logische oder narrative Abfolge zwischen Erzähleinheiten, d.h. die eigentliche Erzählstruktur modelliert wird. So folgen bspw. bei einem animierten Multimedia-Comic die einzelnen Szenen einer Episode narrativ aufeinander.*
- *Simultan-Beziehungen zur Modellierung narrativer Gleichzeitigkeit von Erzähleinheiten. Mit Hilfe einer solchen Beziehung kann z.B. der simultane Ablauf zweier Handlungsstränge ausgedrückt werden.*
- *Voraussetzung-für-Beziehungen, mit denen sich modellieren läßt, daß eine Erzähleinheit Voraussetzung für eine oder mehrere andere Erzähleinheiten ist. Bspw. kann eine Rückblende Voraussetzung für das Verständnis nachfolgender Handlungsstränge sein.*

Darüber hinaus gibt es eine *Ist-Mitspieler-Beziehung*, die zwischen Erzähleinheiten und den Objekten der anderen Arten von Klassen bestehen kann. Sie zielt auf die an einer Erzähleinheit beteiligten Objekte ab. So kann bspw. eine Szene in einem bestimmten Raum ablaufen, in dem dann gewisse Agenten auftreten.

Klassen für Objekte mit räumlicher Ausdehnung (S-Klassen[3]). In HMA spielen vielfach Objekte eine Rolle, die in erster Linie durch ihre Räumlichkeit charakterisiert sind. Dies gilt typischerweise für Spiele, animierte interaktive Comics oder virtuelle Museen. Man denke bspw. an die in solchen Applikationen oftmals auftretenden Räume und die darin befindlichen Gegenstände. Zur Modellierung von zwei- und dreidimensionalen Objekten verfügt *HyDev*s Domainmodell über S-Klassen. Diese können über diverse räumliche Beziehungen wie *Grenzt-an-Beziehungen* oder *Enthalten-in-Beziehungen* miteinander verbunden sein.
Mitunter weisen räumliche Objekte ein spezielles dynamisches Verhalten auf. Dieses ist den mit Erzähleinheiten modellierten Abläufen strukturell sehr ähnlich. Zur Erfassung einer Verhaltenseigenschaft räumlicher Objekte kann die zugehörige S-Klasse daher in einer *Beschreibt-Verhalten-von-Beziehung* zu einer N-Klasse stehen. Wenig komplexe Verhaltenseigenschaften können in Analogie zu Attributen unmittelbar im Rahmen der S-Klasse selbst erfaßt werden. Zum Beispiel könnte ein elektronischer Produktkatalog für Feuerwerkskörper die einzelnen Produkte in Form von Animationen illustrieren. Das zugehörige Domainmodell enthielte dann eine S-Klasse Feuerwerkskörper, die solche Verhaltenseigenschaften wie Flugverhalten oder Knall- und Lichteffekte beschreibt.

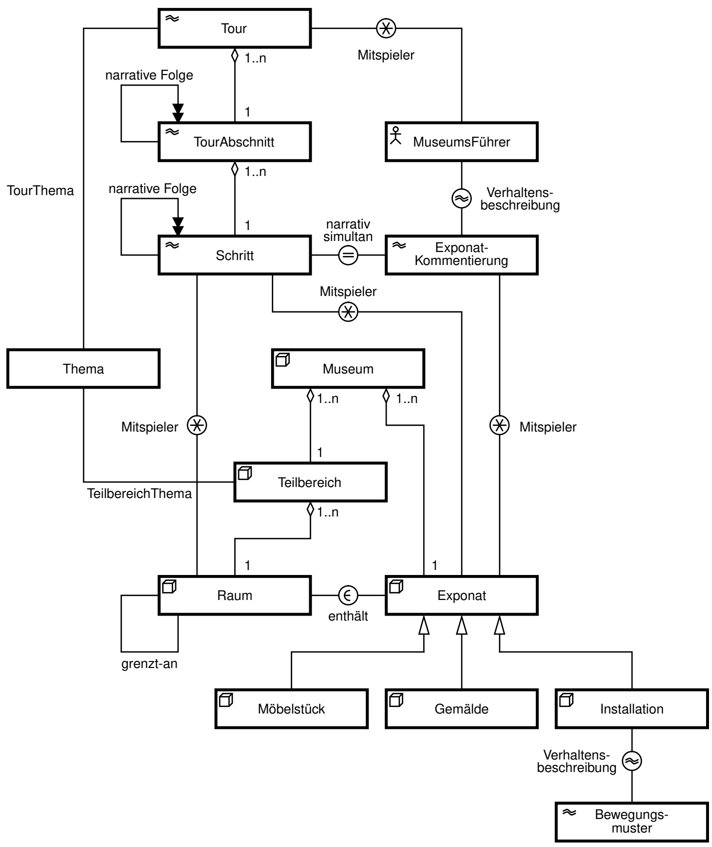
Klassen für Agenten (A-Klassen). Eine dritte große Gruppe für HMA typischer Objekte sind *Agenten*. Diese werden mit A-Klassen modelliert und treten als Mitspieler in Erzähleinheiten auf. Auch Agenten können Verhaltenseigenschaften aufweisen, die wie bei räumlichen Objekten modelliert werden. Im Unterschied zu räumlichen Objekten sind solche Verhaltenseigenschaften jedoch Ausdruck einer gewissen Eigenständigkeit. So können Agenten bspw. bestimmte Aufgaben haben, auf Veränderungen anderer Objekte reagieren oder (in Grenzen) eigene Ziele verfolgen. Zudem sind Agenten in der Lage, untereinander zu agieren. Klassische Agenten sind z.B. die Charaktere von Abenteuerspielen.

3.Das S steht für Spatial.

Beispiel

Als Beispiel sei ein als HMA realisiertes virtuelles Kunstmuseum betrachtet. Das Museum ist räumlich in Teilbereiche aufgeteilt, die sich jeweils auf ein bestimmtes Thema beziehen. Ein Teilbereich setzt sich aus einzelnen, aneinander grenzenden Räumen zusammen, die die Exponate des Museums enthalten. Bei den Exponaten werden Möbelstücke, Gemälde und Installationen unterschieden. Bei Installationen bewegen sich Objekte in komplexer Weise. Ein Benutzer kann Touren durch Teilbereiche des Museums machen. Eine solche Tour hat ein bestimmtes Thema und setzt sich aus aufeinanderfolgenden Abschnitten zusammen, die ihrerseits aus einzelnen Schritten bestehen. Touren werden von einem Museumsführer geleitet, der durch die einzelnen Räume wandert und die darin befindlichen Exponate erläutert.

Die folgende Abbildung zeigt einen Ausschnitt aus dem Domainmodell zu einer solchen HMA. Auf die Angabe von Attributen, Operationen und Verhaltenseigenschaften mußte dabei aus Platzgründen verzichtet werden.



Die Notation des Domainmodells lehnt sich an UML an. Demzufolge werden Assoziationen über einfache Verbindungslinien zwischen Klassen, Ganzes-Teile-Beziehungen und Vererbungsbeziehungen durch Linien mit Rauten bzw. Pfeilspitzen gekennzeichnet.
Damit die diversen Arten von Klassen voneinander unterschieden werden können, erhalten ihre Symbole in der linken oberen Ecke jeweils ein kleines Signet. N-Klassen besitzen eine doppelte Schlangenlinie, S-Klassen einen dreidimensionalen Würfel und A-Klassen ein stilisiertes Männchen. Die Symbole für konventionelle Klassen erhalten kein Signet.
Das Museum, seine Teilbereiche und Räume sowie die Exponate sind räumlichen Objekte. Folglich werden sie mit Hilfe von S-Klassen modelliert. Da Möbelstükke, Gemälde und Installationen spezielle Exponate sind, wird zwischen den entsprechenden Klassen eine Vererbungsbeziehung eingetragen. Daß Räume aneinandergrenzen und Exponate enthalten, wird durch eine Grenzt-an-Beziehung bzw. eine Enthält-Beziehung zwischen den entsprechenden Klassen ausgedrückt.
Der Museumsführer wird als typischer Agent mittels einer A-Klasse erfaßt. Sein komplexes Verhalten wird mittels einer eignen N-Klasse spezifiziert. Die Tour und ihre Bestandteile sind Erzähleinheiten, die als N-Klassen Eingang in das Modell finden. Die narrative Gleichzeitigkeit zwischen einem Schritt als Teil eines Tourabschnittes und dem Verhalten des Museumsführers bei der Kommentierung von Exponaten wird durch eine Simultan-Beziehung Rechnung getragen. Mit Hilfe von Mitspieler-Beziehungen wird modelliert, welche Objekte an den Erzähleinheiten partizipieren. Das Thema der Tour schließlich wird mittels einer konventionellen Klasse spezifiziert.

2.3 Instanzenmodell

Das Instanzenmodell setzt sich aus Instanzen der im Domainmodell erfaßten Klassen zusammen. Für jedes zur Laufzeit auftretende Objekt enthält das Instanzenmodell eine Modellkomponente. Mit Ausnahme der Attribute für komplexe Medienobjekte (Filme, Animationen, Video, Bilddaten usw.) werden für alle Attribute deren Werte spezifiziert. Die Instanzen sind über instantiierte Beziehungen des Domainmodells miteinander verbunden.

Ein Instanzenmodell kann auch Objekte enthalten, die nicht Ausprägung einer bestimmten Klasse aus dem Domainmodell sind. Bei solchen Objekten handelt es sich um Kollektionen von Instanzen. Derartige Kollektionen können durchaus heterogen sein. Auf diese Weise werden bei der Erstellung des Instanzenmodells in gewisser Weise Schwerpunkte gesetzt. Es entsteht somit nicht grundsätzlich allein durch einfache, quasi mechanische Instantiierung des Domainmodells.

2.4 Repräsentationsmodell

Das Repräsentationsmodell knüpft an das Instanzenmodell an und modelliert Aspekte der Objektrepräsentation und Benutzerinteraktion. Berücksichtigung finden dabei bewußt nur die wichtigsten strukturellen Gesichtspunkte, um das Repräsentationsmodell nicht mit zu diesem Zeitpunkt zweitrangigen Details zu überladen. Im einzelnen wird spezifiziert,

- *wie ein Objekt repräsentiert wird. Dabei wird in erster Linie der Medienobjekttyp (Text, Graphik, Image, Video, Animation usw.) angegeben. Auf Festlegungen von Formaten (GIF, MPEG usw.) oder gar Namen von Dateien mit Medienobjekten wird verzichtet.*
- *wie eine Objektrepräsentation aus anderen Repräsentationen aufgebaut ist*
- *ob mehrere Objekte eine gemeinsame Repräsentation besitzen*
- *welche räumlichen und zeitlichen Beziehungen zwischen den Repräsentationen bestehen*
- *welches Ausgabemedium (Fenster, Fensterteilbereich, Audiokanal, externes Ausgabegerät usw.) für die Wiedergabe einer Repräsentation verwendet wird*
- *welche Kommandos der Benutzer an einer Repräsentation auslösen kann*
- *welche Interaktionstechniken das Auslösen eines bestimmten Kommandos erlauben*
- *wie die Reaktion auf Events (z.B.: eine Objektrepräsentation mit zeitlicher Ausdehnung erreicht ihr Ende) und Kommandos aussehen*
- *wie die Navigationsstruktur aufgebaut ist*

3 Aktueller Stand und Ausblick

In der Vergangenheit haben wir existierende Produkte verschiedener Kategorien genauer untersucht. *HyDev* ist aus den bei diesen Reengineering-Projekten gewonnenen Erkenntnisse heraus entwickelt. Momentan arbeiten wir am Repräsentationsmodell, für das wir insbesondere eine praktikable Notation entwerfen. Parallel dazu untersuchen wir weitere Fallstudien, mit denen wir die Tragfähigkeit des Ansatzes validieren wollen. In Zukunft werden wir u.a. Regeln für die Konsistenz zwischen den drei Modellen erarbeiten. Des weiteren interessiert uns, wie die mit *HyDev* gewonnenen Modellierungen auf Praxis-Standards wie MHEG, HyTime oder das Lingo-Format (Director) abgebildet werden können. Eine andere interessante Fragestellung ist, ob und wie aus den *HyDev*-Modellen eine konventionelle Flowchart- und Timeline-Darstellung gewinnbar ist. Schließlich werden wir ein Edittool entwickeln, welches die Erstellung der *HyDev*-Modelle unterstützt. Dazu werden wir unseren generischen Modelleditor *GenTool* anpassen, der im Zusammenhang mit dem an unserem Lehrstuhl entwickelten FLUID-Ansatz entstanden ist [6,7].

Literatur

1. Schwabe, D. und Rossi, G., *The Object-Oriented Hypermedia Design Model*, Communications of the ACM, Vol. 38, No. 8, S. 45-46, August 1995
2. Kozel, K., *The Interactive Killing Fields*, Multimedia Producer, No. 5, 1996 http://www.kipinet.com/mmp/mmp_may96/feat_killing.html
3. Isakowitz, T., Stohr, E. A. und Balasubramanian, P., *RMM: A Methodology for Structured Hypermedia Design*, Communications of the ACM, Vol. 38, No. 8, S. 34-43, August 1995
4. Garzotto, F., Paolini, P. und Schwabe, D., *HDM - A model-based approach to hypermedia applications design*, ACM Trans. Info. Syst. 11, 1, S. 1-23, Januar 1993
5. Gaulding, J. und Katz, B., *Hypermedia application design: A structured approach*, Designing User Interfaces for Hypermedia, W. Schuler, J. Hannimann und N. Streitz, Eds. Springer-Verlag, ESPRIT series, Heidelberg, S. 43-52, 1994
6. Homrighausen, A. und Voss, J., *Tool support for the model-based development of interactive applications - The Fluid approach*, Proceedings of Eurographics Workshop on Design, Specification, Verification of Interactive Systems DSV-IS ’97, Eurographics Series, 1997
7. Kösters, G., Six, H.-W., Voss, J., *Combined Analysis of User Interface and Domain Requirements,* Proc. 2nd IEEE Int. Conf. on Requirements Engineering, Colorado Springs, 1996