

Jürgen Hauser (IPVR)
Frank Scholze (UB Stuttgart)

Integration und Verarbeitung multimedialer Dokumente in Digitalen Bibliotheken - der MAVA-Ansatz

Anforderungen an multimediale Dokumentensysteme

Die Bedeutung multimedialer Dokumente in Forschung und Lehre muss heute kaum noch ausführlich begründet werden, sie ist vielfach festgestellt worden.¹ Ihre Vorteile bei der Wissensdarstellung und -vermittlung sind evident, wenn ihre Erstellung und Vermittlung im geeigneten Kontext geschieht. Dieser Kontext ist je nach Anwendungsgebiet unterschiedlich, jedoch lassen sich einige grundlegende Anforderungen an multimediale Dokumente bzw. Dokumentensysteme beschreiben.

Aus Sicht eines potentiellen Autors bzw. Autorenteam multimedialer Dokumente ist die Relation zwischen dem benötigten Aufwand beim Erstellen und dem Nutzen bzw. Mehrwert des veröffentlichten Dokuments von entscheidender Bedeutung. Daraus resultieren einige grundlegende Voraussetzungen:

- Das multimediale Autorensystem muss einfach und schnell zu erlernen und so weit wie möglich intuitiv bedienbar sein, so dass der bislang hohe Aufwand beim Erarbeiten einer spezifischen Programmiersprache verringert wird.
- Die Bereitstellung eines grafischen Editors sollte es ermöglichen, auch anspruchsvolle multimediale Dokumente per „drag & drop“ zu erstellen. Dies schließt die Komposition zeitlicher, räumlicher und funktionaler Beziehungen von Medienelementen (Text, Bild, Video, Audio etc.) mit ein.
- Bereits erstellte Teile von Dokumenten mit spezifischer Anwendungsfunktionalität (z.B. Konzepte für die Spezifikation von Fragen in einem computergestützten Training oder eine geographische Positionsänderung als alternative Interaktionsform) sollten technisch unbeschränkt wieder verwendbar sein, d.h. die gekapselte Anwendungsfunktionalität kann auch von anderen Autoren mit geringem Aufwand genutzt werden. Dazu sollte das Autorensysteme eine geeignete Abstraktion (das Dokumentenmodell) für die Kapselung bereitstellen.
- Das Autorensystem sollte das gerade bearbeitete spezifische Anwendungsgebiet unterstützen bzw. möglichst einfach um die jeweils benötigte Funktionalität erweitert werden können.
- Das Präsentationssystem stellt minimale Anforderungen an ein Benutzersystem für die Wiedergabe von Multimedia-Dokumenten (Beispielsweise die Java-Runtime in einem Web-Browser).
- Dynamische Erweiterbarkeit sollte die nachträgliche automatische Anpassung, d.h. die Spezialisierung für konkrete Anwendungsgebiete, der Funktionalität des Präsentationssystems ermöglichen

Die Wiederverwendbarkeit spielt eine umso bedeutendere Rolle, je mehr spezifische und heterogene Anwendungsgebiete erschlossen wurden. Im Gegensatz zu den bisherigen Lösungen ist ein Autorensystem, das die oben genannten Forderungen erfüllt, universell einsetzbar. Von einer weitaus größeren Verbreitung und Akzeptanz im Hochschulbereich kann mit hoher Wahrscheinlichkeit ausgegangen werden. An den Hochschulen sind heterogene, gleichzeitig jedoch hochspezialisierte Anwendungsgebiete gegeben, es fehlt in vielen Bereichen jedoch an Personal,

Mitteln und Möglichkeiten, um die komplexen bzw. teuren existierenden multimedialen Autorensysteme einsetzen zu können.

Wissenschaftliche Bibliotheken und Rechen- bzw. Medienzentren stellen zusätzliche Anforderungen an ein multimediales Dokumentensystem. Sie beziehen sich in besonderem Maße auf die mögliche Integration multimedialer Dokumente in bestehende Systeme, ihre Bereitstellung und Archivierung.

Eine Kernforderung in Zeiten knapper Finanz- und Personalressourcen ist die Integration von Nachweis und Bereitstellung multimedialer Dokumente in bereits bestehende Informationssysteme, die mit möglichst geringem Aufwand durchführbar sein soll. Dies ist primär durch die Berücksichtigung der vorhandenen (nur z.T. normierten) Protokolle und Schnittstellen zu gewährleisten. Das Präsentationssystem muss möglichst flexibel sein hinsichtlich der Erweiterung um neue Aufgabenbereiche. Medienelemente in neuen bzw. weiterentwickelten Formaten sollten hinzugefügt werden können, ohne das Präsentationssystem als Ganzes wechseln zu müssen und ohne dem Benutzer Installationsaufwand zu verursachen. Offene Standards sollten so weit wie möglich berücksichtigt werden. Bei der Archivierung von multimedialen Dokumenten sollten möglichst „flache“, d.h. hard- und softwareunabhängig strukturierte, Dateiformate Verwendung finden, wie z.B. SGML oder XML. Diese Dateiformate können bei Bedarf mit einfachen Editoren und Viewern bearbeitet und betrachtet werden. Das Programmformat der zu archivierenden multimedialen Dokumente muss plattformunabhängig sein. Der Zugriff auf multimediale Dokumente soll für den Benutzer der Bibliothek - ebenso wie die Archivierung für die Bibliothek selbst - weitgehend hard- und softwareunabhängig bzw. plattformunabhängig möglich sein.

Die Plattformunabhängigkeit der Dokumente und des Präsentationssystems stellt eine unabdingbare Voraussetzung für eine funktionierende Langzeitverfügbarkeit von multimedialen Dokumenten dar, wenn man davon ausgeht, dass der Archivierungsansatz Technikumuseum finanziell und organisatorisch nicht flächendeckend durchführbar ist und die Konzepte Emulation und Konvertierung langfristig nicht verlustfrei arbeiten können.

Der MAVA-Ansatz

Die Universitätsbibliothek Stuttgart und das Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR) der Universität Stuttgart entwickeln im Rahmen des von der DFG geförderten Schwerpunktprogramms „Verteilte Verarbeitung und Vermittlung digitaler Dokumente“ (V2D3)² das erweiterbare multimediale Dokumentensystem „Multimedia Document Versatile Architecture“ (MAVA)³, das die oben beschriebenen Anforderungen erfüllt.

Grundlage des erweiterbaren Multimediasystems MAVA sind das Dokumentenmodell und das Präsentationssystem. Beide unterstützen die dynamische Erweiterbarkeit. Erweiterbarkeit bedeutet im Zusammenhang mit MAVA, dass eine Unterstützung für die Erstellung multimedialer Dokumente aus bestimmten Anwendungsgebieten möglich ist. Unterschiedliche Anwendungsgebiete bedingen unterschiedliche Konzepte für die Beschreibung von Beziehungen der Medien untereinander. In diesem Artikel gehen wir von einer ganz allgemeinen Beschreibung von Dokumenten aus, in der ein Dokument aus Medien und Beziehung zwischen Medien besteht.

Einfache statische (zeitunabhängige) Textdokumente lassen sich damit ebenso beschreiben wie multimediale (zeitabhängige und aus mehreren unterschiedlichen Medientypen bestehende) Dokumente. Ein Textdokument besteht meist aus Text- und Bildmedien, die in einer räumlichen Beziehung stehen. Die Zeitabhängigkeit multimedialer Dokumente drückt sich dadurch aus, dass bestimmte Informationen dem Benutzer erst zu einem vorher festgelegten Zeitpunkt präsentiert werden. Auch wenn Multimedia-Dokumente klassische Textdokumente umfassen können, sollte je nach Verwendungszweck des Dokuments das passende Format und Werkzeug zur Erstellung gewählt werden, d.h. MAVAs will kein Ersatz für existierende Textverarbeitungen o.ä. sein.

Bei der Entwicklung eines erweiterbaren Multimediasystems ist zu beachten, dass auch in Zukunft noch neue Konzepte integriert werden können, an die bei der ursprünglichen Entwicklung nicht gedacht wurde oder nicht gedacht werden konnte, wie z.B. neue Konzepte aus dem Bereich computergestütztes Lernen und Lehren oder multimediale Spiele. Spiele als multimediale Dokumente umfassen komplizierte zeitliche und räumliche Beziehungen einzelner Elemente. Diese Beziehungen werden in MAVAs durch Operatoren beschrieben. Beispielsweise kann in einem Labyrinth die Beziehung beschrieben werden, dass die Spielfigur von einer gegnerischen Figur verfolgt werden soll. Diese Beziehung ist eine Kombination einer räumlichen (Figur bewegen) und einer zeitlichen (in einem gewissen Tempo, damit es für den Spieler fair bleibt) Beziehung.

Dokumentenmodell

Das Dokumentenmodell beschreibt den Aufbau eines MAVAs-Dokumentes. Dem MAVAs-Dokumentenmodell liegt das im folgenden beschriebene Meta-Dokumentenmodell zu Grunde. Hauptelemente sind die Medienelemente und die Operatoren. Ein Medienelement steht dabei für atomare Medientypen wie beispielsweise ein Video oder ein Bild. Der Begriff Medienelement soll hier sehr allgemein gefasst sein, so kann er beispielsweise ebenso für chemische Moleküle oder Texteingabefelder stehen. Ein Operator wird dazu verwendet, um bestimmte Beziehungen zwischen Medienelementen zu beschreiben. Ein Operator hat in diesem Sinne eine ausschließlich beschreibende Funktionalität, die Semantik einer Menge von inhaltlich zusammengehörigen Operatoren wird durch eine sogenannte Managerkomponente bei der Präsentation umgesetzt. Beispiele für Operatoren sind einfache temporale oder räumliche Beziehungen. Operatoren können aber auch komplexe Beziehungen beschreiben, wie z.B. den Fortgang der Präsentation in Abhängigkeit von der Beantwortung einer Frage.

Der Ausschnitt aus einem Beispieldokument in Abbildung 1 verdeutlicht die Anwendung des Dokumentenmodells. Das Beispieldokument besteht aus vier Operatoren und fünf Medienelementen.

Die Medienelemente sind zum einen ein Molekül, zwei Selektionsfelder und zwei Texte. Der Ausschnitt stellt einen Container dar, der die Medien und einen Operator enthält. Im MAVAs-Modell können Container, die beliebige Medienelemente und ihre Verknüpfungen enthalten, zur Strukturierung von Dokumenten verwendet werden. So stellt der abgebildete Container einen logischen Teil, in diesem Fall eine Frage einer Lerneinheit, des gesamten Dokuments dar.

Für den strukturellen Aufbau werden drei Klassen von Operatoren verwendet. Dies sind zeitliche, räumliche und Frage-Spezifikations Operatoren. Die zwei räumlichen Operatoren sind der *above*-Operator und der *left*-Operator. Sie bestimmen, dass das Molekül sich räumlich über (above) dem ersten Selektionsfeld befindet und das erste Selektionsfeld sich links vom erläuternden Text. Der *solution*-Operator spezifiziert die richtige Lösung der Frage, in diesem Beispiel muss also das erste Selektionsfeld ausgewählt sein.

Die Operatoren zur Beschreibung des zeitlichen Ablaufs wurden hier der Einfachheit halber weggelassen, um die Übersicht zu behalten bietet der Editor einen Filtermechanismus an um bestimmte Operatorklassen auszublenden. Für eine Frage ist erwünscht, dass alle Medienelemente gleichzeitig auch gleich lange angezeigt werden. Dies könnte beispielsweise durch zeitliche Intervalle (hier die Dauer der jeweiligen Präsentation) ausgedrückt werden.⁴

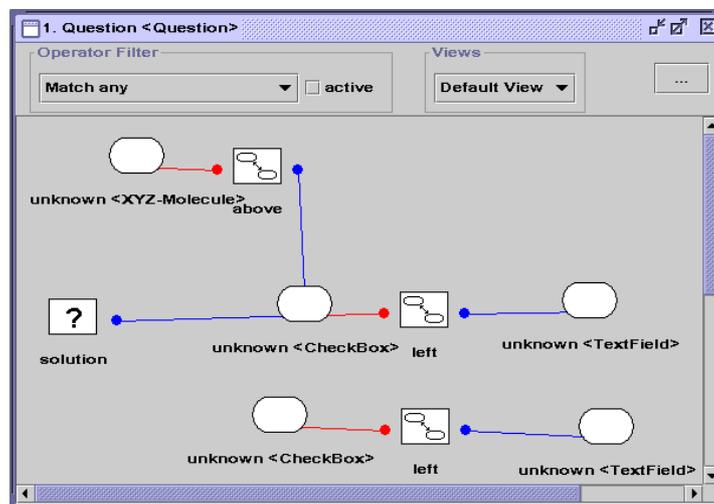


Abbildung 1: Ausschnitt aus einem Beispieldokument

Architektur des Präsentationssystems

Im vorherigen Abschnitt wurde das Dokumentenmodell beschrieben. Es handelte sich dabei um ein Meta-Dokumentenmodell. Konkrete Dokumente aus bestimmten Anwendungsgebieten lassen sich dadurch beschreiben, dass spezifische Konzepte benutzt werden. Diese Konzepte bestehen aus einer Reihe von Operatoren, wie z.B. das Konzept der temporalen Intervall-Operatoren oder Operatoren um Fragen und Antworten zu spezifizieren. Beim Erstellen eines Dokumentes können dann genau diese Operatoren verwendet werden, wobei in ein Dokument wiederum beliebig viele Konzepte integriert werden können. Um konsistente Dokumente zu erhalten, muss natürlich geprüft werden, ob die sogenannten Manager-Komponenten, oder kurz Manager, in dieser Kombination erlaubt sind, und im negativen Fall die Verwendung unterbunden werden. Die Semantik der Operatoren muss für die Präsentation des Dokuments an irgendeiner Stelle realisiert sein. Dies geschieht in den Managern. Damit ist jeder Operator einem Manager zugeordnet. Ähnlich wie das Dokumentenmodell um neue Operatoren erweiterbar ist, so ist auch die Architektur des Präsentationssystems um neue Manager erweiterbar. Vor der Präsentation eines Dokumentes werden alle benötigten Manager nachgeladen. Dieser Prozess geschieht automatisch ohne Eingreifen des Benutzers, der Prozess kann deshalb als ad-hoc Erweiterbarkeit bezeichnet werden.

Die Architektur des Präsentationssystems ist in Abbildung 2 dargestellt. Dabei ist zu erkennen, dass die Manager auf der MAVA-Engine aufbauen. Die MAVA-Engine übernimmt Grundfunktionalitäten, wie das Laden der Dokumente und Manager und die Verwaltung der Klassen zur Realisierung der dynamischen Erweiterbarkeit. Sie bietet aber auch Unterstützung für die Entwicklung neuer Manager. Häufig wiederkehrende Aufgaben bei der Entwicklung (z.B. Abwicklung der Kommunikation zwischen Manager und Medienelementen) werden von der MAVA-Engine übernommen, was den Erstellungsprozess vereinfacht. Vor allem zu Beginn einer breiteren Nutzung von MAVA müssen noch häufiger Manager entwickelt werden, um neue Konzepte zu integrieren.

Die Architektur umfasst außer Operatoren, Managern und der MAVA-Engine noch Medien und Viewer. Beide sind für die Realisierung der Semantik von Medienelementen verantwortlich. Die Medien beschreiben die Eigenschaften eines Mediums innerhalb eines Dokumentes (z.B. Größe oder Verweis auf die Mediendaten durch einen URL). Der Viewer ist für die Präsentation der Medien verantwortlich. So lassen sich zum Beispiel die unterschiedlichen Graphikformate von Bildern als ein Medium „Bild“ und mehrere Viewer für unterschiedliche Graphikformate (GIF, JPEG, PNG,...) zusammenfassen. An dieser Stelle sieht man, dass nicht nur neue Konzepte integriert werden können, sondern auch neue Medien und Viewer, falls neue Formate entwickelt werden. Beispielsweise kann ein Medienelement „Molekül“ und die zugehörigen Viewer für unterschiedliche Darstellungen des Moleküls (z.B. 3D-Ansicht) integriert werden. Damit lassen sich auch Medienelemente, die spezifisch für ein Anwendungsgebiet sind, integrieren. Diese Informationen können beim Erzeugen der Metadaten wiederverwendet werden.

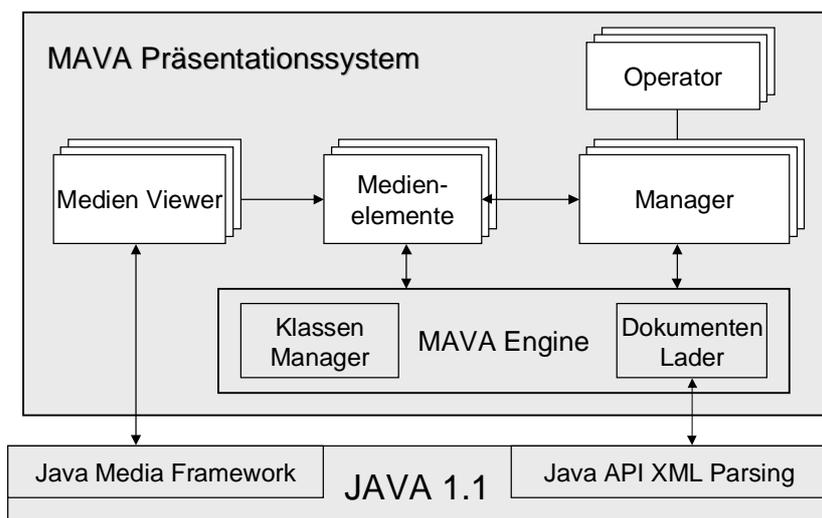


Abbildung 2: Architektur des MAVA-Präsentationssystems

Die unterste Schicht der Architektur ist nicht direkter Bestandteil von MAVA. Sie soll nur verdeutlichen, dass MAVA auf der Programmiersprache Java und zwei speziellen APIs (application programming interface zu deutsch Anwendungsprogrammierschnittstelle) basiert. Die speziellen APIs sind das Java Media Framework (JMF) und das Java API for XML Parsing (JAXP). Das JMF erlaubt

die einfache Verarbeitung von kontinuierlichen Medien in Java, die für die Präsentation von Video- und Audiosequenzen benötigt wird. Das JAXP erlaubt das Einlesen und Schreiben von XML-Dateien. Die herausragende Eigenschaft von Java ist die Plattformunabhängigkeit, welche Java zur „Internetsprache“ werden lässt. Plattformunabhängigkeit bedeutet, dass die Programme auf den unterschiedlichsten Plattformen (z.B. verschiedene Unix-Dialekte oder Windows NT) ablauffähig sind. In der Praxis gibt es dabei die kleine Einschränkung, dass eine Laufzeitumgebung (genannt Virtuelle Maschine) für diese Plattformen vorhanden sein muss.

In diesem Artikel kann nicht auf alle technischen Einzelheiten des MAVA-Ansatzes eingegangen werden.⁵ Ein technischer Aspekt, die Speicherung von Dokumenten, ist für digitale Bibliotheken jedoch von besonderem Interesse und wird daher im nächsten Abschnitt noch etwas eingehender betrachtet.

Speicherung von MAVA-Dokumenten

Das Speichern von MAVA-Dokumenten ist durch die Abbildung des Dokumentenmodells auf XML⁶ im Prinzip einfach zu erreichen. Damit baut das Dokumentenformat von MAVA auf einem Standard für den Dokumenten- und Datenaustausch im Internet auf. XML ist eine Meta-Sprache, in der sich beliebige Markup-Sprachen beschreiben lassen. Markup bedeutet, dass Teile eines textuellen Dokuments durch sogenannte Tags markiert werden. Diese Tags können zusätzliche Meta-Informationen über den markierten Bereich enthalten.

Die tatsächliche Realisierung der Speicherung ist naturgemäß etwas komplexer. Eine Besonderheit von MAVA-Dokumenten ist die Tatsache, dass zu einem Dokument auch die benötigten Manager gehören. Um Dokumente austauschen zu können, müssen auch Manager ausgetauscht werden. Die Manager sind aber technisch gesehen Klassen einer Programmiersprache. Deshalb müssen bei der Speicherung von MAVA-Dokumenten auch die Manager berücksichtigt werden. Die Manager sind jedoch, ebenso wie das Dokument selbst, plattformunabhängig und können somit auch auf den unterschiedlichsten Plattformen verwendet werden.

Damit die richtigen Manager geladen werden, müssen in das Dokumentenspeicherformat zusätzliche Informationen integriert werden. Diese Informationen beschreiben, welche Manager für die Präsentation benötigt werden und wo sie gegebenenfalls zu finden sind. Neben der Information, welche Manager zu einem Dokument gehören, werden weitere Informationen für den Aufbau der internen Darstellung eines Dokuments im Präsentationssystem benötigt. Die durch die Container-Elemente eingeführte Enthaltenseins-Beziehung von Medienelementen und Operatoren kann direkt auf die Schachtelung von XML-Elementen abgebildet werden. Die Informationen, aus welchen Objekten die Datenstruktur der internen Darstellung aufgebaut ist, müssen zusätzlich bereitgestellt werden. Es werden daher spezielle MAVA-DTDs (document type definitions) erzeugt. Da jedes Dokument unterschiedliche Medienelemente und Manager verwendet, werden individuell für jedes Dokument neue DTDs erzeugt. Aufgrund der Erweiterbarkeit ist es nicht möglich, „die“ MAVA-DTD anzugeben, da diese alle (auch zukünftige) Erweiterungen umfassen müsste.

Integration verschiedener Arten von Metadaten im MAVA-Editor

Der beschriebene Ansatz von MAVA verbindet technische Metadaten, welche z.B. die Beziehungen von Medienelementen beschreiben, mit inhaltsbeschreibenden bzw. „bibliographischen“ Metadaten (Autor, Titel, Schlagwörter, Klassifikationen etc.)⁷ Sie werden gemeinsam in XML-Dokumenten gespeichert und die jeweiligen Anteile bei Bedarf von unterschiedlichen Anwendungen verarbeitet.

Die Bearbeitung der Dokumente durch den Autor erfolgt mittels einer graphischen Benutzungsoberfläche in einem spezifischen MAVA-Editor (genannt MED). Eine plattformunabhängige Realisierung des Editors ist aus den selben Gründen wie für das Präsentationssystem sinnvoll, um nämlich die unterschiedlichen im Hochschulbereich verwendeten Plattformen abdecken zu können. Eine Ausrichtung auf eine einzige Plattform wie beispielsweise in Firmen ist hier nicht abzusehen, auch wenn dies bisweilen postuliert wird. Um die Erstellung von Dokumenten mit MED so einfach wie möglich zu halten, kann fast das komplette Dokument unter Verwendung der Maus als Eingabegerät gestaltet werden. Medienelemente und Operatoren können durch "drag & drop" einem Container hinzugefügt werden. Lediglich die Eingabe numerischer bzw. textueller Parameter zu einzelnen Medienelementen sowie die Erfassung der inhaltsbeschreibenden Metadaten Autor, Titel, Schlagwörter etc muss mittels Tastatur vorgenommen werden. Ein entscheidender Vorteil liegt darin, dass für die Erstellung eines Dokuments keine Programmiersprache und zugehörige Programmierkenntnisse notwendig sind.

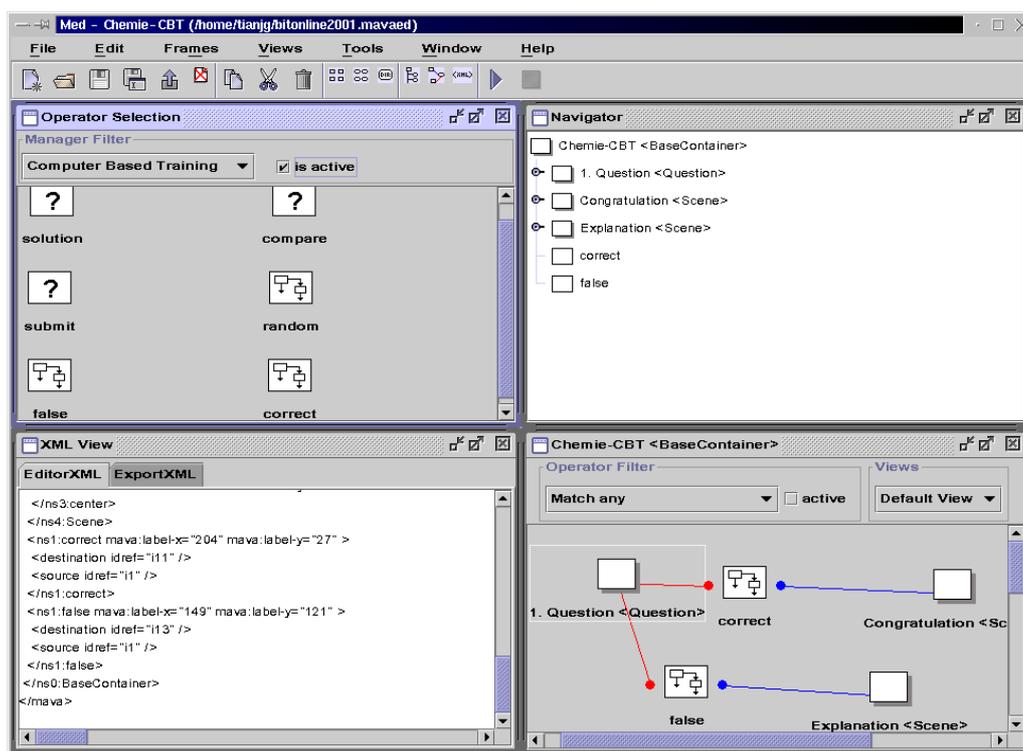


Abbildung 3: Überblick über den MAVA-Editor

Mit dem Editor können so komplexe multimediale Dokumente im Sinn der Definition von Steinmetz, "die in mindestens einem kontinuierlichen (zeitabhängigen) und einem diskreten (zeitunabhängigen) Medium kodiert sind"⁸ ohne Programmierkenntnisse bearbeitet werden. Die Erstellung und Bereitstellung ähnlich komplexer Dokumente wie sie MAVA bietet, basieren heute in der Regel auf

proprietärer Hard- und Software wie z.B. MILESS in Essen.⁹ Trotzdem ist auch hier auf Benutzerseite je nach den verwendeten Medienelementen eine Reihe verschiedener Viewer nötig, die separat installiert werden müssen. Einer der wenigen uns bekannten Ansätze, der inhaltsbeschreibende Dublin Core (DC) Metadaten für multimediale Dokumente benutzt und diese Dokumente auch unter einer vollständig in einen Standard-Webbrowser integrierten Oberfläche anbietet ist PATRON¹⁰ (Performing Arts Teaching Resources Online). Der funktionale Ansatz der Multimedia-Dokumente ist allerdings auf computerbasiertes Lernen beschränkt und auf Benutzerseite werden verschiedene Plug-Ins benötigt, so dass die Integration in den Webbrowser nur unter Windows-Betriebssystemen möglich ist.

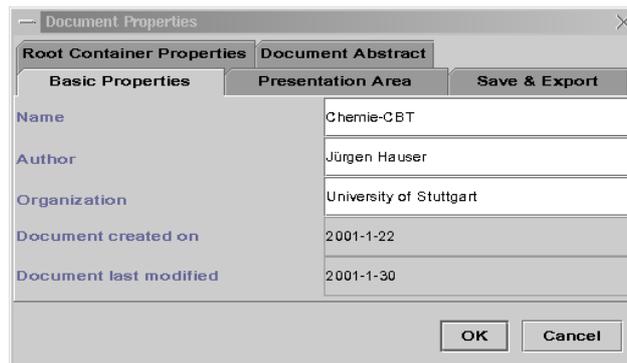


Abbildung 4: Der Meta-Datendialog für ein Dokument

Als Richtlinie für die inhaltsbeschreibenden Metadaten in MAVA wurde die XML-Repräsentation des DImeta-Datenmodells¹¹ herangezogen. Dieses Datenmodell entstand in Baden-Württemberg durch die Zusammenarbeit verschiedener Projekte im Bereich digitaler Bibliotheken. Als wesentlich sind dabei einmal die Projekte zum Einsatz des Produktes IBM Digital Library zu nennen. Der zweite Hauptansatz ging aus von der Zusammenarbeit des Bibliotheksservice-Zentrums Baden-Württemberg (BSZ)¹² und der Universität Stuttgart (OPUS)¹³ bei der Anwendung von Dublin Core Metadaten für den Austausch bibliographischer Daten elektronischer Hochschulschriften. Hier wurden die fortschreitenden Ergebnisse der Dublin Core Initiative¹⁴ mit bewährten bibliothekarischen Standards verbunden. Rasch wurde jedoch klar, dass beim Einsatz technisch unterschiedlicher Applikationen ein gemeinsames Datenmodell die entscheidende Brücke darstellt. Deshalb entschied man sich, das DImeta-Modell für den Einsatz innerhalb der Digitalen Bibliothek Baden Württemberg¹⁵ verbindlich zu empfehlen.

MAVA profitiert von dieser Entscheidung, indem inhaltsbeschreibende Metadaten im XML-Format, die anhand der DImeta-DTD validiert werden können, in das MAVA-Dokument eingebettet werden. Abbildung 4 zeigt einen Dialog des Autorenwerkzeugs zur Eingabe von Metadaten, wie beispielsweise den Autor. Nebenbei pflegt das Autorensystem zusätzliche Informationen wie das Erstellungsdatum oder das Datum der letzten Änderung. Ebenso kann ein Abstract eingegeben werden. Zusätzlich unterstützt der MAVA-Ansatz die automatische Erzeugung von inhaltsbeschreibenden Metadaten. Dazu werden die Informationen, welche Medienelemente und Manager in einem Dokument verwendet werden herangezogen. So lässt sich beispielsweise aus der Verwendung von Medienelementen der Chemie und eines Managers für computergestütztes Lernen schließen, dass es sich bei diesem Dokument um ein Lerndokument für

Chemiestudenten handelt. Einzelne Medienelemente bzw. Manager können ohne weiteres z.B. auf Begriffe der Schlagwortnormdatei (SWD) oder Notationen von Klassifikationen abgebildet und dem Editor zur Verfügung gestellt werden. Eine derart automatisierte Inhaltsbeschreibung wurde angedacht, da die in MAVAs verwendeten Medienelemente und Manager teilweise spezifische Anwendungsfunktionalität besitzen, diese jedoch nicht so fein ausdifferenziert ist, so dass gerade ein universelles Beschreibungsinstrument wie die SWD geeignet erscheint.¹⁶ Der folgende Ausschnitt eines MAVA-Dokuments kann dies verdeutlichen:

<pre><?xml version="1.0" encoding="UTF-8" ?> <mava xmlns:dmeta="http://www.dlmeta.de/ dlmeta/2000/Dlmeta.dtd"> <dmeta:object> <Title><TitleMain>Chemical CBT</TitleMain> </Title> <Creator> <Person> <CompleteName FirstName="Jürgen" LastName="Hauser"> </CompleteName> </Person> </Creator> <Description> <Abstract> ... gekürzt ... </Abstract> </Description> <Date> <Created>2001-01-22</Created> <Issued>2001-01-31</Issued> </Date> <SubjectScheme="SWD">Computeruntersütztes Lernen</Subject> <SubjectScheme="SWD">Chemie</Subject> </dmeta:object> <requiredManager ns="de.uni-stuttgart.ipvr.v.s" package="cvt.jar" /> <requiredMediaItem ns="de.uni-stuttgart.ipvr.v.s" package="molecule.jar" /></pre>	<pre><BaseContainer> <Question name="1. Question" id="i0"> <molecule src="../molecules/benzene.mol"> ... </Question> <Scene name="Congratulation" id="i1"> ... </Scene> <Scene name="Explanation" id="i2"> ... </Scene> <correct> <source idref="i0"> <destination idref="i1"> </correct> <false> <source idref="i0"> <destination idref="i1"> </false> </BaseContainer> </mava></pre>
---	--

Abbildung 5: Ausschnitt aus der XML-Repräsentation eines MAVA-Dokuments

Das MAVA-Präsentationssystem überliert die inhaltsbeschreibenden Informationen zwischen `<dmeta:object>` und `</dmeta:object>` als nicht relevant für die unmittelbare Darstellung der einzelnen Medienelemente und ihrer Beziehungen. Die Kennzeichnung relevanter Teile erfolgt dabei durch die Verwendung von XML-Namensräumen. Ein System wie OPUS kann jedoch ausschließlich diesen Teil des Dokuments lesen und trägt bei Bedarf die entsprechenden Angaben in seine DC-basierte Datenbank ein. Aus diesen Angaben wird dann wie für andere Dokumente auch eine Indexseite generiert, über die der Zugriff auf das Dokument selbst möglich ist. Zudem können die Daten mit weiteren Nachweissystemen, wie der SWB-Verbunddatenbank oder dem Online Katalog der Region Stuttgart (BISSCat) ausgetauscht werden.¹⁷ Auch hierbei erweist sich die Verwendung von Begriffen der SWD zur Beschreibung von MAVA-Dokumenten als integrativer Ansatz, da die SWD in diesen Datenbanken das einheitlich verwendete Referenzsystem der inhaltlichen Erschließung darstellt.

Zusammenfassung

MAVA unterscheidet sich damit in doppelter Hinsicht von den meisten Ansätzen im Bereich der Multimedia-Bereitstellung und Archivierung. Zum einen wird die längerfristige Bereitstellung durch ein plattformunabhängiges Präsentationssystem (Java) und Dokumentenformat (XML) realisiert. Es können komplexe Dokumente erzeugt und verarbeitet werden, die den Anforderungen aus unterschiedlichsten Anwendungsbereichen entsprechen und nicht nur Sammlungen meist homogener Medienelemente wie Audio oder Video. Die Erweiterbarkeit sichert zusätzlich den Fall ab, dass ein neues Anwendungsgebiet erschlossen werden soll. Für nicht erweiterbare Ansätze würde dies im schlimmsten Fall die Integration eines neuen Systems bedeuten.

Zum anderen ist die Integration verschiedener Typen von Metadaten in einer XML-Repräsentation realisiert, so dass in einem Dokument nicht nur die Semantik der verschiedenen zu seiner Präsentation nötigen Komponenten (Manager, Operatoren) beschrieben wird, sondern auch so weit wie möglich die Daten, die zum textuellen Retrieval dieses Dokuments dienen können. Dadurch ist nicht nur ein direkter Suchzugriff auf MAVA-Dokumente möglich, sondern auch ein Austausch dieser Daten mit Nachweissystemen wie OPUS oder dem Verbundkatalog des Südwestdeutschen Bibliotheksverbundes.

¹Vgl. u.a. Bund-Länder-Kommission für Bildungsplanung und Forschungsförderung: *Multimedia in der Hochschule*. Bonn, 2000 <<http://www.blk-bonn.de/papers/heft85.pdf>> Deutscher Wissenschaftsrat: *Empfehlungen zur Hochschulentwicklung durch Multimedia in Studium und Lehre*. Mainz, 1998. <<http://www.wrat.de/drucksachen/drs3536-98/drs3536-98.htm>>

²<<http://www.cg.cs.tu-bs.de/dfgspp.VVVDD>>

³<<http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/mava/>>

⁴Eine ausführliche Beschreibung dieser temporalen Operatoren findet sich in: Thomas Wahl, Kurt Rothermel: Representing Time in Multimedia Systems, in: *IEEE 1. International Conference on Multimedia Computing and Systems*, Boston, 1994, S. 538-543.

⁵Technische Informationen zu MAVA finden sich in: Jürgen Hauser, Kurt Rothermel: Spezifikation and Realization of an Extensible Multimedia System, in: Hans Scholten, Marten J. van Sinderen (Eds.): *Proceedings of the 7th Interactive Distributed Multimedia Systems and Telecommunication Services Workshop (IDMS)*, Berlin, 2000, LNCS 1905.

⁶Tim Bray, Jean Paoli, C.M. Sperberg-McQueen (Eds.): *Extensible Mark-up Language (XML) 1.0, Working Recommendation*, W3C, 1998.

⁷Zur Typologie vgl. *Introduction to Metadata*, Los Angeles, 1998, S. 17 bzw. S. Boll et al.: Overview on using Metadata to manage Multimedia Data, In: *Multimedia Data Management*, New York, 1998, S. 12ff.

⁸Ralf Steinmetz: *Multimedia-Technologie*, 3. Aufl. Berlin, 2000, S. 19.

⁹Beate Tröger: Multimedia in Forschung und Lehre: Das Beispiel MILESS Essen, In: *BIT Online* 4, 1999 <<http://www.b-i-t-online.de/archiv/1999-04/fach1.htm>> Kap. 2.3.

¹⁰<<http://www.lib.surrey.ac.uk/patron2/>>

¹¹<<http://www.dlmeta.de>>

¹²<<http://www.bsz-bw.de/diglib/medserv/>>

¹³<<http://elib.uni-stuttgart.de>>

¹⁴<<http://www.purl.org/DC/>>.

¹⁵Christoph-Hubert Schütte: Die Digitale Bibliothek Baden-Württemberg, In: *BIT Online* 3, 2000, S. 309-312 <<http://www.b-i-t-online.de/archiv/2000-03/fach1.htm>>

¹⁶Eine - allerdings manuelle - Erschließung von Multimedia-Dokumenten mittels Begriffen der SWD ist beispielsweise auch im Rahmen der Virtuellen Hochschule Oberrhein (VIROR) vorgesehen. <<http://www.viror.de>>

¹⁷Eine ausführlichere Beschreibung von OPUS und des Datenaustausches mit dem SWB findet sich in: Frank Scholze: "Einbindung elektronischer Hochschulschriften in den Verbundkontext am Beispiel OPUS" In: Beate Tröger (Hrsg.), *Wissenschaft online: Elektronisches Publizieren in Bibliothek und Hochschule*. Frankfurt, 2000, S. 406-420.