

# **Ein Metriken basierter Ansatz für das Informationsmanagement von eLearning Projekten**

Uwe Blazey – Reiner Dumke

UBISNET - Dorfstr.1 - 29386 Oberholz / Wierstorf - uwe@blazey.de

Otto-von-Guericke-Universität Magdeburg

Softwaretechnik - Institut für Verteilte Systeme – dumke@ivs.cs.uni-magdeburg.de

**Abstract:** Um ein erfolgreiches eLearning Projektmanagement leisten zu können müssen Informationen über abgeschlossene, laufende und zukünftige Projekt-Aktivitäten bekannt sein und jederzeit in aktueller, übersichtlicher Form der Projektleitung zur Verfügung stehen. Im Folgenden werden die bisherigen Arbeiten, die im Zusammenhang mit dieser Zielsetzung stehen, in einer Übersicht vorgestellt. Dabei basieren alle Untersuchungen auf Daten und projektrelevanten Informationen von UBISNET. Ziel ist es, zukünftig die Projektrisiken weiter zu minimieren, Prozesse zu optimieren, dabei die bisherige Qualität beizubehalten und wo es möglich ist Verbesserungen herbeizuführen. Erfolg versprechend sind diese Arbeiten nur, wenn hierbei relevante und messbare Ergebnisse berechnet werden, auf deren Basis es möglich ist, in komplexe Prozesse korrigierend eingreifen zu können. Die hier vorgestellten Ansätze sind Teil des kontinuierlichen Qualitäts- und Optimierungsmanagement von UBISNET und stehen im Zusammenhang mit Arbeiten zum Thema *Software-Messungen und -Bewertungen* der Fachabteilung Softwaretechnik, Institut für Verteilte Systeme an der Otto-von-Guericke- Universität Magdeburg.

## **1 Grundkonzepte und Komponenten von eLearning-Systemen**

Multimedia Projekte und speziell eLearning Projekte sind sehr aufwendig und kostenintensiv. Es ist daher wichtig, die möglichen Risiken in jeder Projektphase abschätzen und eingrenzen zu können. Dazu bedarf es kontinuierlicher Analysen, Bewertungen, um gegebenenfalls an kritischen Stellen die richtigen Veränderungen und Korrekturen vornehmen zu können. Informationen, Ergebnisse und Erfahrungen aus bereits abgeschlossenen Projekten können dazu beitragen, Abschätzungen, Prozesse und Produkte kontinuierlich zu verbessern. Sowohl die Kosten als auch die Entwicklungszeiten können so bei gleich bleibenden Anforderungen und Zielsetzungen optimiert und reduziert werden. Damit entsprechende Analysen und Bewertungen vorgenommen werden können, müssen alle Informationen funktionell geordnet und klassifiziert werden, so dass sich Zusammenhänge und Beziehungen zwischen den verschiedenen Informationsgruppen bestimmen lassen. Die Bestimmung von Zusammenhängen und Beziehungen kann dabei zunächst formell durchgeführt werden, um diese mathematisch korrekt zu erfassen. Diese Vorgehensweise erleichtert die Umsetzung dieser Konzeption in eine Software-Anwendung für das Informationsmanagement [Zu97] [ED04]. Im Folgenden wird ansatzweise aufgezeigt, wie

Gruppierungen und Klassifizierungen von eLearning Projektinformationen vorgenommen und die Zusammenhänge und Beziehungen formell beschrieben werden können. Die daraus resultierenden Ergebnisse können dazu verwendet werden, ein umfassendes Informationssystem für das eLearning Projektmanagement zu entwickeln [Ec04]. Die dem Informationssystem zugrunde liegende Konzeption ermöglicht es, neben der Verwaltung und Auswertung softwaretechnisch relevanter Daten und Informationen, Funktionen zu integrieren, die es ermöglichen, Teile des Informationsmanagement zu automatisieren.

UBISNET beschäftigt sich mit der Entwicklung von informationstechnischen eLearning Anwendungen und unterliegt dabei den fortlaufenden Veränderungen bei den Technologien und den damit verbundenen Informationen. Aus diesen Gegebenheiten heraus gehört die kontinuierliche Suche nach Konzepten und Lösungen, unter Berücksichtigung der speziellen, internen Infrastruktur, mit zu den primären Aufgabenstellungen. Die hier vorgestellten Ansätze können ohne weiteres für allgemeine eLearning Projekte, sowohl im "Kleinen", als im "Großen", Verwendung finden. An dieser Stelle sei angemerkt, dass es durchaus erhebliche Unterschiede in der Sicht auf eLearning Projekte geben kann. (Unternehmenssicht, universitäre Sicht, etc.). Die GI-Fachgruppe Metriken; Software-Messung und -Bewertung beschäftigt sich bereits seit vielen Jahren mit entsprechenden Konzepten, Modellen und Methoden. Die hier zugrunde liegenden Lösungsansätze und Erfahrungen gilt es, soweit möglich, auf eLearning Projekte abzubilden, beziehungsweise an den Anforderungen des eLearnings anzupassen und anzuwenden.

### **1.1 eLearning Projektmodell**

Das Thema eLearning umspannt viele verschiedene Fachgebiete. Daher ist es für spezielle fachspezifische Untersuchungen nützlich, eine Trennung der mit dem eLearning verbundenen Fachbereiche vorzunehmen. Auf Basis des Projekt-Modells, das für eLearning-Projekte spezialisiert und erweitert wurde, wird eine solche Abgrenzung vorgenommen. Unter dem Begriff "Informationstechnik für das Lernen" - kurz IT4L - wird eine Abgrenzung für das softwaretechnische Fachgebiet definiert.

#### *Definition:*

*Unter dem Begriff IT4Learning, kurz IT4L, sind alle informationstechnischen Produkte, Prozesse und Ressourcen zusammengefasst, die für das Lernen unter Verwendung von Informations- und Kommunikationstechnik, benötigt werden.*

Im eLearning Projektmodell, siehe unter Abbildung 1, werden die softwaretechnisch relevanten Aspekte durch die beiden Felder *Prozess* und *Ressourcen* abgedeckt. Fachübergreifende Themen und Aspekte sind dem Bereich *Produkt* zugeordnet. Im Vordergrund dieser Darstellung stehen die jeweiligen Sichten auf die eLearning Thematik. Zum einen, die Sicht auf die eLearning Anwendungsentwicklung, dann die Sicht auf die eLearning-Tool Entwicklung und schließlich eine Vielzahl anderer Sichten, wie beispielsweise die von Anwendern, Didaktikern, Soziologen auf das Produkt und die damit verbundenen Möglichkeiten, etc.

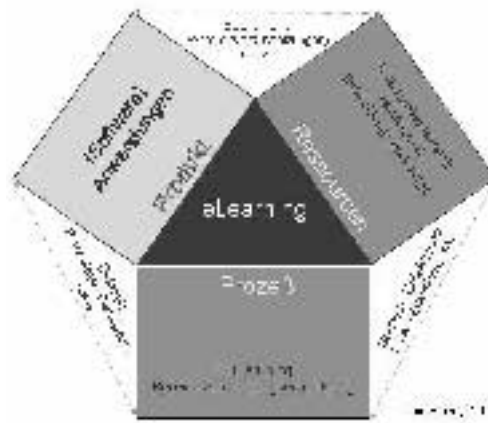


Abbildung 1: Das auf eLearning-Projekte abgestimmte Projekt-Modell

Die Strichlinienpfeilverbindungen deuten dabei an, dass es zu den einzelnen Abgrenzungen Schnittstellen gibt oder geben kann, die zu berücksichtigende, fachübergreifende Aspekte beinhalten können. Beispielsweise sind Vorgaben aus anderen Fachdisziplinen denkbar, die es softwaretechnisch umzusetzen gilt. Ein Vorgehen in umgekehrter Richtung ist ebenfalls möglich.

## 1.2 Daten für das eLearning

Einer der bekanntesten Ansätze, Zusammenhänge von eLearning Inhalten und die damit verbundenen Daten darzustellen, ist der von Hodgins (siehe Abbildung 2).

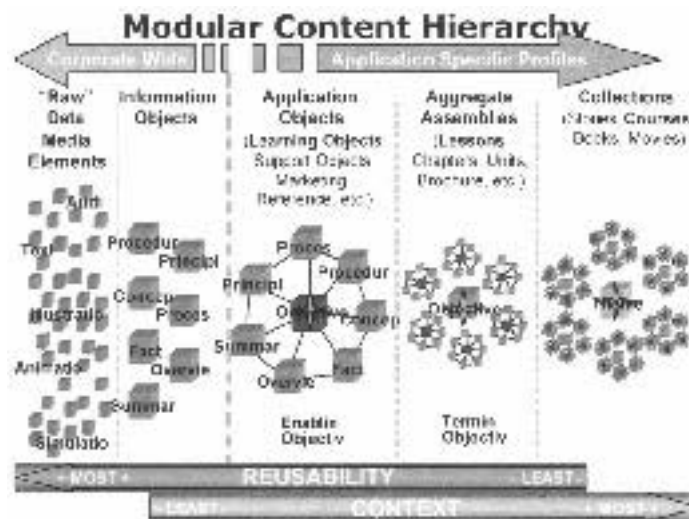


Abbildung 2: Beschreibung der eLearning Inhalte nach Hodgins 2002

Der in dieser Abbildung dargestellten Hierarchie sind Pfeile zugeordnet, über die sich Aussagen über den Grad der möglichen Wiederverwendbarkeit und Inhaltsfülle der Daten aus den einzelnen Gruppierungen ablesen lassen. Um weitere, detaillierte und aussagekräftige Informationen zu erhalten, bedarf es tief greifender Untersuchungen und Klassifizierungen.

### 1.3 eLearning -Informationsmodell

In vielen Fachartikeln zum Thema eLearning wird publiziert, dass es vorteilhaft sei, eLearning Anwendungen modular in kleine Lernabschnitte aufzubauen oder überhaupt nur kleine Anwendungen zu entwickeln, die in etwa 20 Minuten Lehrstoffvermittlung bieten. Auch aus softwaretechnischer Sicht ist es vorteilhaft kleine Module, die möglicherweise wieder verwendet werden können, zu entwickeln, die dann zusammengesetzt, eLearning-Anwendungen ergeben. In Sachen Software-Test und Software-Wartung sind ebenfalls Vorteile gegenüber Spaghetti-Code Programmen zu erwarten. Bei ständig wechselndem und zu überarbeitendem Anwendungsdesign kann eine modular aufgebaute Anwendung Nachteile in Sachen Anpassungsaufwand bedeuten, wenn dieser Prozess nicht automatisiert werden kann. Die Frage, die sich an dieser Stelle ergibt, ist, aus welchen Daten-Teilen eLearning-Anwendungen bestehen oder bestehen können und auf welche Basis sich diese Daten-Teile beziehen. Im Folgenden werden alle Daten, die eine eLearning-Anwendung ausmachen und nutzbar machen, ihrer Funktionsweise nach klassifiziert, gruppiert und beschrieben.

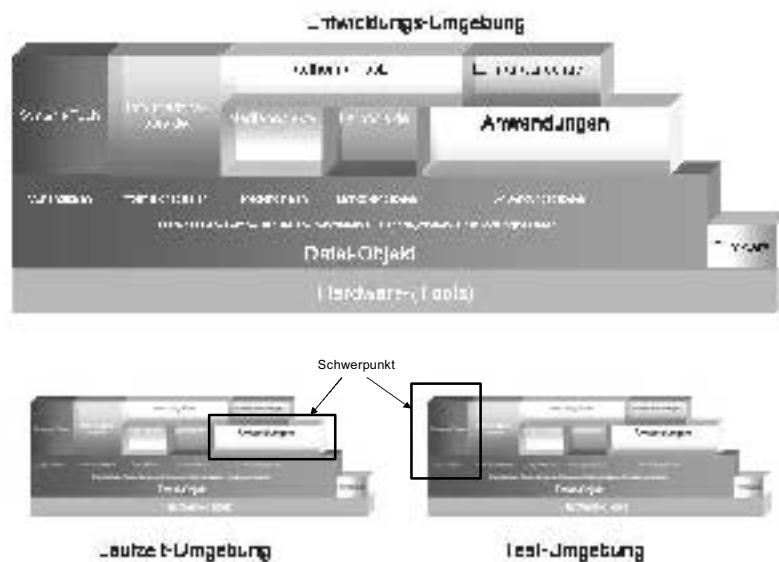


Abbildung 3: eLearning-Informationsschichtenmodell

Abbildung 3 zeigt ein Schichtenmodell für eLearning Umgebungen. In großer Darstellung ist zunächst der Aufbau einer eLearning Entwicklungsumgebung zu

erkennen. Zum Entwicklungsprozess gehört aber auch eine Test-Umgebung, die hier dem gleichen Schichtenmodell entspricht, real aber zur Entwicklungsumgebung auf verschiedene Systemkonfigurationen und Hardwarevoraussetzungen basiert. Hierbei überwiegen die System-Tools und Hardware-Tools. Eine reine Laufzeit-Umgebung dagegen wird stärker auf den Anwendungsteil ausgelegt sein. Anwender bewegen sich überwiegend auf der oberen Informationsschicht, Test-Ingenieure zudem auf der mittleren und unteren Schicht. Grundsätzlich ist davon auszugehen, dass für jeden dieser Informationsbausteine, beziehungsweise Informationsgruppen, spezielles Wissen erforderlich ist. Selbst Anwender benötigen grundlegende Kenntnisse für die Bedienung und Nutzung von eLearning-Anwendungen. Im Folgenden wird festgelegt, was unter den folgenden Begriffen zu verstehen ist und wie diese Begriffe einzuordnen sind.

*Definition Informationsobjekte (IO):*

**Informationsobjekte** sind elektronisch gespeicherte Daten, die Rohfassungen von Lehrstoffen, Vorgehensweisen, Konzepte, Modelle, etc. für die Entwicklung von eLearning Anwendungen enthalten.

*Definition Medienobjekte (MO):*

**Medienobjekte** sind in Dateien gespeicherte Informationen in Form von Texten, Grafiken, Präsentationen, Audios, Videos oder ausführbaren Software-Programmen.

*Definition Lernobjekte (LO):*

**Lernobjekte** sind Software-Module, die unter Verwendung von Medienobjekten, ein zielgerichtetes, möglichst interaktives und multimediales Lernszenario darstellen.

*Definition Lernanwendungen (LA):*

**Lernanwendungen** sind einsetzbare eLearning Programme, die Lernobjekte, Medienobjekte und eigene Programm-Routinen enthalten, um in sich abgeschlossene, zielgerichtete Schulungen, anhand von aufeinander aufbauenden Lernszenarien, bereitzustellen.

Für die Konzeption, Entwicklung, Test, Wartung und Verwendung von eLearning Anwendungen werden Werkzeuge (Tools) benötigt. Je nach Verwendungszweck können folgende Tools unterschieden werden:

*Definition Authoring-Tools (AT):*

**Authoring-Tools** (Autoren-Werkzeuge) sind Software-Programme für die Erstellung und Generierung von Informationsobjekten, Medienobjekten, Lern-Objekten und Lern-Anwendungen. Sie generieren Mediendateien, Anwendungsdateien und Quellcodedateien. Mitunter stellen sie den Anwendungen benötigte Systemdateien zur Verfügung, damit die Anwendungen auf entsprechenden Systemen ausführbar sind.

*Definition System-Tools (ST):*

**System-Tools** bilden die Basis für Ausführbarkeit und Anwendung von Medien-Objekten, Lern-Objekten, Lern-Anwendungen, Authoring-Tools und Hardware-Tools. Sie bestehen aus System-Programmen, Treibern, Systemdateien und Systemwerkzeugen.

Definition Hardware-Tools (HW):

**Hardware-Tools** sind (Standard) Computer- und Peripherie-Ausstattungen die als Basis für die Entwicklung, Tests und sicheren Ausführung von eLearning Anwendungen benötigt werden.

Aus dieser Definition lassen sich Hardware-Komponenten weiter klassifizieren, beispielsweise in **Systemhardware, Equipment und Lern-Hardware**.

Definition Datei-Objekte (DO):

**Datei-Objekte** sind alle Dateien, die einer der oben definierten Objekte, Anwendungen und Tools zugeordnet werden können, zum Entwicklungsprozess von eLearnig - Anwendungen gehören und zu dessen Ausführung benötigt werden.

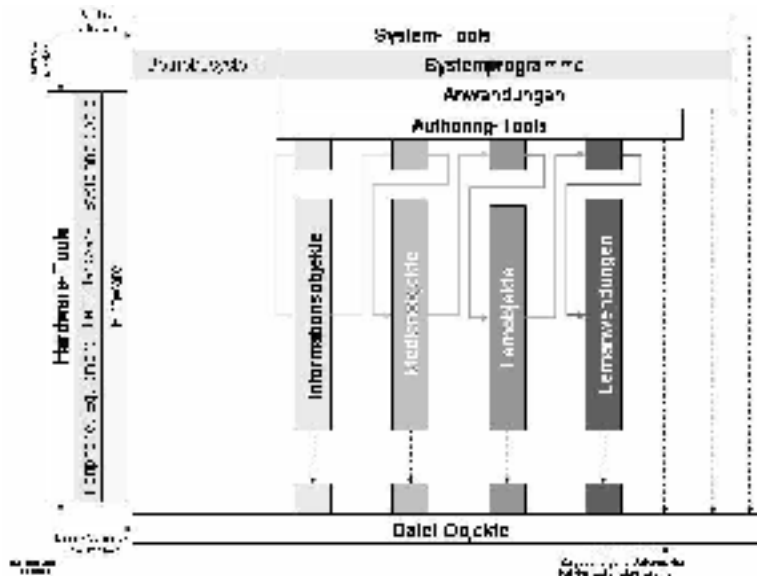


Abbildung 4: Informationsschema einer eLearning-Entwicklungsumgebung

Im Gegensatz zum Schichtenmodell zeigt Abbildung 4 schematisch die Informationsgruppen mit Ihren Schnittstellen zueinander für eine eLearning Entwicklungsumgebung. Tief greifendere Zusammenhänge können hieraus nur erahnt werden. Dies ist Thema des folgenden Abschnitts.

## 2 Informationsmanagement von eLearning-Systemen

Die in Abschnitt 1 hergeleitete verbesserte Informationsgranularität ermöglicht eine detaillierte Untersuchung der einzelnen Informationsgruppen unter Berücksichtigung von Beziehungen und Abhängigkeiten untereinander. In diesem Abschnitt werden unter anderem Ansätze für ein Vorgehen bei der Informationsanalyse aufgezeigt. Die Komplexität des gesamten softwaretechnischen Informationsgehaltes wird dadurch in

kleinere, überschaubarere Komplexitätsgruppen zerlegt. Durch entsprechende Konzeptionen und Modellierungen kann aus den dabei resultierenden Erkenntnissen heraus die Verwaltung der Informationen als Datenbankanwendung erfolgen. Durch geschickte Zusammenstellung und Aufbereitung von Informationsobjekten lassen sich Informationen zu Medienobjekten, Lernobjekten und Lernanwendungen automatisiert generieren. Dies ist ein Ansatz für eine automatisierte Metadatengenerierung. In Sachen Skalierbarkeit von Lernobjekten und Lernanwendungen können verwendete Authoring- und System-Tools oder Dateiformate sofort Aufschluss über kompatible Lernumgebungen geben. Über Beziehungen von Attributen kann der Grad der wieder verwendeten Objekte und von Source-Code ermittelt werden. Solche und ähnliche Zusammenhänge können Grundlage von Optimierungen und Aufwandsabschätzungen sein.

## 2.1 Analyse-Ansätze

In Abbildung 5 wird ansatzweise aufgezeigt, welches Wissen im Unternehmen für die Entwicklung von eLearning Anwendungen relevant ist, verwaltet und organisiert werden muss und somit Gegenstand für Informationsobjekte ist. Die Art und Weise der Aufbereitung von Informationsobjekten ist dabei entscheidend für eine ansatzweise automatisierte Metadatengenerierung. Informationsobjekte gibt es für verschiedenste Aufgabenbereiche. Sie können Anweisungen und Vorgaben sein, die den eLearning-Entwicklungsprozess betreffen. Für Lernobjekte und Lernanwendungen dienen sie als Gebrauchsanleitung oder Beschreibung und können beispielsweise als XML-Dateien vorliegen.

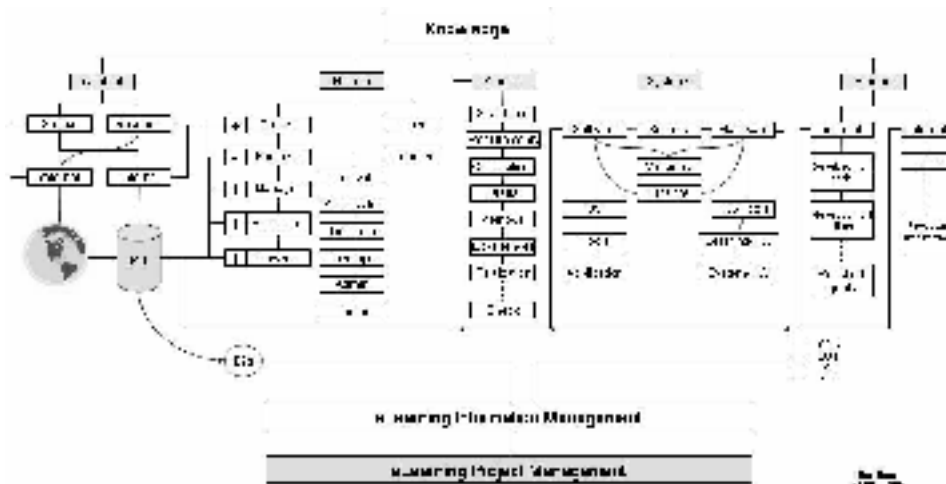


Abbildung 5: eLearning Projekt Wissensbasis für Informationsobjekte

In wie weit die Medienobjekte, Lernobjekte und Lernanwendungen von den Vorgaben der Informationsobjekte im Entwicklungszyklus beeinflusst werden, muss über bestehende Beziehungen untereinander analysiert werden.

## 2.2 Komplexitätsarten und ihre Merkmale

Das Problem im Umgang mit eLearning Projektinformationen liegt in der damit verbundenen, kontinuierlich wachsenden Komplexität. Je mehr Informationen bewältigt werden müssen, desto schwieriger wird es, den Überblick zu behalten. Die dauerhafte Beherrschbarkeit der Komplexität ist daher das primäre Ziel [DB99]. Daher muss analysiert werden, welche Schwierigkeiten im Umgang mit Informationen und deren Verwaltung bestehen und wodurch sie ausgelöst werden. Hierzu dient der folgende Ansatz einer speziell auf eLearning Projekte ausgerichtete Komplexitätsanalyse.

*Definition:*

*Komplexität ist die Gesamtheit aller Merkmale eines Begriffes, Zustandes oder Prozesses*

Diese Definition ist dabei auf die Merkmale, Begriffe, Zustände und Prozesse der softwaretechnischen Aspekte von eLearning Projekten beschränkt. Die bisherigen Gruppierungen und Klassifizierungen beinhalten keine Informationen über Prozesse. Daher kann die Komplexitätsuntersuchung von Prozessen zunächst zurückgestellt werden. Die Komplexitätsuntersuchung kann, wie hier bereits vorgenommen, auf spezielle Aspekte ausgerichtet werden. Eine weitere Spezialisierung ist also möglich und könnte sich auf die Analyse der im Abschnitt 1 vorgenommenen Gruppierungen beschränken. Im Abschnitt 3 wird dies beispielhaft auf Medienobjekte formell angewandt. In der folgenden Tabelle wird aufgezeigt, wie einige relevante Komplexitätsarten und die damit verbundenen Merkmale beschrieben können.

Begriff/ Zustand	Komplexität	Merkmale
Information	Informationskomplexität	Aufwand, Menge, Aktualität, etc.
Daten	Datenkomplexität	Menge
Struktur	Strukturelle Komplexität	Beziehungen, Abhängigkeiten, etc.
Ressourcen	Ressourcen Komplexität	Kompliziertheit (Vielfalt)

Tabelle 1: Auswahl von Komplexitäten

Auf weitere Erklärungen und Beschreibungen zu dieser Thematik wird verzichtet, da dies den Rahmen dieser Dokumentation sprengen würde.

## 3 Metriken für das Informationsmanagement

Medien-Objekte lassen sich in Einzel-Objekte und Objekt-Kompositionen unterscheiden. Eine Bilddatei beispielsweise ist ein Einzelmedium. Eine Flash Datei kann dagegen Bilder, Audiodateien und Programm-Code enthalten, ist also eine Komposition verschiedener Medienobjekte.



### 3.1 Komplexitätsmetriken

In diesem Abschnitt werden erste Komplexitätsmetriken, zunächst nur auf die Medienobjekte beschränkt, vorgestellt.

a) Komposition von Medienobjekt-Strukturen

$$MO = \bigcup_{i=1}^n \{MO_i\}; \quad \forall i, n \in \mathbb{N}$$

Für  $n=1$  ist MO ein Einzel-Objekt und für  $n>1$  eine Komposition von Objekten. In [DK98] wird die Komplexität für ein Programm wie folgt festgelegt:

"Die *Rechnerische Komplexität* (computational complexity) beschreibt die Kompliziertheit der Implementation eines Algorithmus und beispielsweise die zu erwartende Laufzeit"

"Die *Psychologische Komplexität* (psychological complexity) beschreibt die Schwierigkeit der Arbeit mit dem Programm (als Verständnis des implementierten Algorithmus). Dabei bezieht sich diese Komplexität vor allem auf den Programmquellcode und beeinflusst wesentlich den Aufwand für die Programmierung".

Dieser Ansatz ähnelt dem, wie in 2.2 bereits vorgestellt und wird an dieser Stelle auf die Medienobjekte mit ihren vergleichbaren Problemstellungen angewandt:

b) Die *Datenkomplexität* von Medien-Objekten wird bestimmt durch die Anzahl aller Medien-Objekte des Datenbestandes

$$C_{Dat}(MO) = |MO|$$

c) Die *Informationskomplexität* der verschiedenen Medien-Typen wird bestimmt durch den Aufwand, der bei einer Speicherung und Ausführung anfällt:

$$C_{Exp}(MO) = \sum_{i=1}^n Cap(MO_i); \quad \forall i, n, Cap \in \mathbb{N}, Cap \text{ in Bit}; (Cap = \text{Capacity})$$

d) Die *Ressourcen-Komplexität* der Medienobjekte wird gekennzeichnet durch deren Kompliziertheit. Dies ist abhängig vom verwendeten Autorentool und der Fertigkeit damit umzugehen, wobei ordinal gilt: 1 – sehr einfach bis 10 – sehr schwierig

$$C_R(MO) = \sum_{i=1}^n C(Tool_{MO_i}); \quad \forall i, n \in \mathbb{N}; C(Tool_{MO_i}) = \{1..10\}$$

$$\text{wobei } C_{R_{Median}}(MO) = \frac{\sum_{i=1}^n C(Tool_{MO_i})}{n}; \quad \forall i, n \in \mathbb{N}$$

die durchschnittliche strukturelle Medien-Objekt Komplexität darstellt und einen Wert zwischen 1 und 10 ergibt. In Verbindung mit der Datenkomplexität ist folgende übersichtliche Darstellung beider Komplexitäten möglich:

$$C_{Dat_R}(MO) = \frac{|MO|}{\frac{\sum_{j=1}^m C(Tool_j)}{m}} = C_{Dat}(MO)_{C_{av}(Tool)}$$

Beispiel:  $C_{Dat_R}(MO) = 2356_7$  für 2356 Medien-Objekte mit einer durchschnittlichen strukturellen Komplexität von 7.

e) Die *funktionale Komplexität* spielt nur bei interaktiven Medien-Objekten eine Rolle, die das Nutzerverständnis bei der Bedienung des Medien-Objektes widerspiegelt und wird hier nicht weiter betrachtet.

f) In diesem Bereich fällt auch die so genannte *diagnostische Komplexität*, die im Aufwand in der Fehlersuche besteht. Diese beiden Komplexitätstypen spielen bei den Lern-Objekten eine weit aus größere Rolle. Auch dieser Fall wird hier nicht weiter betrachtet.

g) Medien-Objekte können mehrfache Verwendung finden, beispielsweise wenn sie in verschiedenen Lernobjekten wieder verwendet werden. Dieser Aspekt entspricht der *strukturellen Komplexität*.

Für die Anzahl der Abhängigkeiten zu den verschiedenen Lern-Objekten gilt,

$$C_{LO}(MO) = \frac{\sum_{i=1}^n |LO|}{n} MO_i; \quad \forall i, n \in \mathbb{N}; \text{ (LO = Lern-Objekt)}$$

Für die Anzahl der Abhängigkeiten zu den verschiedenen Lern-Anwendungen gilt,

$$C_{LA}(MO) = \frac{\sum_{i=1}^n |LA|}{n} MO_i; \quad \forall i, n \in \mathbb{N}; \text{ (LA = Lern-Anwendung)}$$

Für die Anzahl der Abhängigkeiten zu den verschiedenen Authoring-Tools gilt,

$$C_{AT}(MO) = \frac{\sum_{i=1}^n |AT|}{n} MO_i; \quad \forall i, n \in \mathbb{N}; \text{ (AT = Authoring-Tool)}$$

Für die Anzahl der Abhängigkeiten zu den verschiedenen System-Tools gilt,

$$C_{ST}(MO) = \frac{\sum_{i=1}^n |ST|}{n} MO_i; \quad \forall i, n \in \mathbb{N}; \text{ (ST = System-Tool)}$$

Weitere Ausführungen würden den Rahmen dieses Vortrages sprengen und können vom Leser nach dem hier vorgestellten Schema weitergeführt werden.

### 3.2 Erste Anwendungsformen

Die Tabelle 2 zeigt die Ergebnisse erster Berechnungen, die auf eine Lernanwendung angewandt wurden:

		1	2	3	4	5	6	7	8
		$MO$	$C_{Dir}(MO)$	$C_{Exp}(MO)$	$C_r(MO)$	$C_{Lo}(MO)$	$C_{LA}(MO)$	$C_{Ar}(MO)$	$C_{ST}(MO)$
A	Ges.	126	126	153.034	126	99	27	126	6
B	Min.	1	126	0,2	45 <sub>1</sub>	1	1	1	1
C	Max.	4	126	7.121,4	34 <sub>7</sub>	1	1	4	1
		Objekte	Objekte	KByte (*)	MO <sub>Grad</sub>	Objekte	Objekte	Objekte	Objekte

Tabelle 2: Ergebnisse der Medienobjekt-Komplexitätsberechnungen einer Beispielanwendung

Zu (\*): Echte Dateigröße – Im Gegensatz zur Dateigröße auf einem Datenträger

Bei den zur Berechnung verwendeten Medienobjekten handelt es sich um Text, Grafik, Flash-Animationen und Videoclips. Die Lernobjekte sind Authorware Anwendungen, die durch ein Hauptprogramm, das ebenfalls mit Authorware entwickelt wurde, gesteuert werden. Dabei werden in Authorware ActiveX Komponenten verwendet, wie beispielsweise der Windows Media-Player und der Internet Explorer. In der Tabelle 2 ist im Feld A-1 die Gesamtzahl der, für die Berechnung zugrunde liegenden Medienobjekte, eingetragen. Die Medienkomposition liegt dabei maximal bei 4 Medienobjekten und minimal bei einem. (Siehe B-1, C-1). In A-2 ist die Gesamtanzahl der zu untersuchenden Medienobjekte eingetragen, hier wurden für die Lernanwendung 126 Medienobjekte gezählt. In B-2 und C-2 wurde dieser Wert übernommen. Wenn zukünftig Berechnungen über eine Vielzahl von Lernanwendungen erfolgen, könnten diese Felder verwendet werden, um Anwendungen mit minimaler und maximaler Medienobjektverwendung herauszustellen. Die Berechnung des Schwierigkeitsgrades für die Entwicklung der Medienobjekte ist in den Feldern B-3 bis C-3 angegeben, wobei A-3 die Gesamtanzahl der untersuchten Objekte angibt. Die Arbeiten mit einem Texteditor wurden dabei mit 1 (sehr einfach) bewertet, der Umgang mit Macromedia Flash und Video-Bearbeitungswerkzeugen mit maximal 7 (erhöhter Schwierigkeitsgrad). Die Dateigrößen für Medienobjekte sind in A-4 bis C-4 eingetragen. A-5 enthält das Ergebnis, dass 99 Medienobjekte auf Lernobjekte verweisen und zwar minimal und maximal eines (B-5, C-5). Dies bedeutet, dass keine Medienobjekte mehrfach in Lernobjekten enthalten sind. Dies ist auch das Ergebnis in Bezug auf Lernanwendungen (A-6, B-6, C-6), wobei hier nur 27 Medienobjekte direkt in die Lernanwendung integriert sind. In Bezug auf die Authoring-Tools, sind alle Medienobjekte mit mindestens einem vorhandenen Werkzeug entwickelt worden (siehe A-7 bis C-7). Maximal verweisen Medienobjekte auf 4 verschiedene Authoring-Tools. Auf System-Tools verweisen nur 6 Medienobjekte und dabei jeweils genau auf eines (siehe A-8, B-8, C-8). Berechnungen, unter Verwendung weiterer Komplexitätsmetriken und auf alle Gruppierungen ausgedehnt, werden weitaus detailliertere und präzisere Ergebnisse liefern. Eine Verwaltung von Projektinformationen durch Datenbankanwendungen ist somit sehr viel versprechend.

## 4 Erfahrungen und Ausblick

Die Umsetzung dieses Konzeptes hat bei UBISNET bisher dazu geführt, dass strukturelle Änderungen in der Datenhaltung, beziehungsweise Datenverwaltung vorgenommen werden konnten, die zu einem verbesserten Umgang mit Informationen und Dateien geführt haben. Die Objektstrukturen ließen sich dabei ohne Probleme auf den bereits vorhandenen Datenbestand übertragen und anwenden. Als Datenbankanwendung *ePISA (eLearning Projekt- Information, -Steuerung und -Analyse)* werden erste Implementationen vorgenommen. Besonderes Interesse gilt hier der automatisierten Metadaten-Erfassung. Derzeit liegen hier die Probleme in der bestehenden Struktur einiger Informationsobjekte. In Sachen Wiederverwendbarkeit hat sich bisher gezeigt, dass die mehrfache Verwendung von Medienobjekten mit dem vorliegenden Datenbestand problematisch ist und generell darauf verzichtet werden kann. Sie können jedoch als Vorlagen für neue Medienobjekte dienen. Dem Begriff *Wiederverwertbarkeit* oder *Recycling* könnten solche Objekte zugeordnet werden. Die Wiederverwendung von Software-Code hat sich hingegen als sehr nützlich erwiesen, obwohl hier bei Versionierungen ebenfalls angepasster oder geänderter Code entsprechend verwaltet werden muss. *Code-Recycling* wäre hier das Stichwort. Derzeit wird nach Möglichkeiten gesucht, neue, auf dem Markt erhältliche und firmenintern genutzte Software-Versionen so aktuell wie möglich zu halten und den damit verbundenen Wartungs- und Testaufwand für den Fall des Einsatzes, beziehungsweise der Nutzung zu berechnen. Die Generierung entsprechender Wartungs- und Testpläne wären eines der daraus resultierenden Anwendungsmöglichkeiten. Wie diesen Ausführungen zu entnehmen ist, geht es neben der Konzeption und formellen Beschreibungen, letztendlich um die Umsetzung dessen in eine praktische Anwendung. Die Entwicklung einer speziellen Informationsdatenbankanwendung ist dabei nur eine Möglichkeit.

## Literaturverzeichnis

- [DB99] Dumke, R.; Blazey, U.; Zbrog, F.: *Analyse der Beherrschbarkeit der Komplexitäts- und Kausalitätsprobleme bei der Jahr-2000-Umstellung* Preprint Nr.8 , Otto-von-Guericke-Universität Magdeburg, 1999
- [DK98] Dumke, R.; Koeppel, R.: *Komplexität bei der Software-Entwicklung und Softwarezuverlässigkeit*. Tagungsband zum Workshop *Software hoher Zuverlässigkeit, Verfügbarkeit, Sicherheit*, DGLR München, 17. Mai 1995, S. 6-1 - 6-22
- [Ec04] Ebert, C.: *Portfolio-Management for Software Projects*. Tagungsband IWSM / Metrikon 2.-5-Nov. 2004, [S.272-279], Shaker Verlag, 2004.
- [ED04] Ebert, C.; Dumke, R; Bundschuh, M.; Schmietendorf, A.: *Best Practices in Software Measurement*. Springer-Verlag, 2004.
- [Ho02] Hodgins, W.: *The Future of Learning Objects – Davos 2002*, <http://services.bepress.com/cgi/viewcontent.cgi?article=1012&context=eci/etechnologies>
- [SG05] Spath, D.; Ganz, W.: *e3 World-Work, learning, performance – Lernen für die Arbeit von morgen*. Fraunhofer IAO, Universum Verlag Wiesbaden, 2005.
- [Zu97] Zuse, Horst; et al: *A Framework of Software Measurement*. Walter de Gruyter, Berlin 1997.