

Nunmehr zum achten Male liegt ein Sammelband zum Workshop „GeNeMe – Gemeinschaften in Neuen Medien“ vor, der Beiträge zu folgenden Themenfeldern enthält:

- Konzepte für GeNeMe (Geschäfts-, Betriebs- und Architektur-Modelle),
- IT-Unterstützung (Portale, Plattformen, Engines) von GeNeMe,
- E-Learning in GeNeMe,
- Wissensmanagement in GeNeMe,
- Anwendungen und Praxisbeispiele von GeNeMe und
- Soziologische, psychologische, personalwirtschaftliche, didaktische und rechtliche Aspekte von GeNeMe.

Sie wurden aus einem breiten Angebot interessanter und qualitativ hochwertiger Beiträge zu dieser Tagung ausgewählt.

Das Interesse am Thema GeNeMe (Virtuelle Unternehmen, Virtuelle Gemeinschaften etc.) und das Diskussionsangebot von Ergebnissen zu diesem Thema sind im Lichte dieser Tagung also ungebrochen und weiterhin sehr groß.

Die thematischen Schwerpunkte entsprechen aktuellen Arbeiten und Fragestellungen in der Forschung wie auch der Praxis. Dabei ist die explizite Diskussion von Geschäfts- und Betreibermodellen für GeNeMe, insbesondere bei der aktuellen gesamtwirtschaftlichen Lage, zeitgemäß und essentiell für ein Bestehen der Konzepte und Anwendungen für und in GeNeMe.

In zunehmendem Maße rücken weiterhin auch Fragen nach den Erfolgsfaktoren und deren Wechselbeziehungen zu soziologischen, psychologischen, personalwirtschaftlichen, didaktischen und rechtlichen Aspekten in den Mittelpunkt. Deshalb wurde hierzu ein entsprechender Schwerpunkt in der Tagung beibehalten.

Konzepte und Anwendungen für GeNeMe bilden entsprechend der Intention der Tagung auch weiterhin den traditionellen Kern und werden dem Anspruch auch in diesem Jahr gerecht.

Die Tagung richtet sich in gleichem Maße an Wissenschaftler wie auch Praktiker, die sich über den aktuellen Stand der Arbeiten auf dem Gebiet der GeNeMe informieren möchten.

Klaus Meißner / Martin Engeliem (Hrsg.)

# Virtuelle Organisation und Neue Medien 2005

Workshop GeNeMe2005  
Gemeinschaften in Neuen Medien

TU Dresden, 6./7.10.2005

## **D.3 Konfigurierbare Softwarekomponenten zur Unterstützung dynamischer Lern- und Arbeitsumgebungen für virtuelle Gemeinschaften**

*Alexander Roth, Thorsten Hampel*  
*Universität Paderborn*

### **1. Einleitung**

Die Kontexte, in denen Mitglieder virtueller Gemeinschaften zusammen lernen und arbeiten, werden von einer Vielzahl von Faktoren beeinflusst: Während heutige Lernumgebungen oftmals mehrere didaktische Konzepte und ihre flexible Kombinierbarkeit (*blended learning*) berücksichtigen müssen (vgl. [Roth & Suhl 05]), treten wechselnde Anforderungen an Arbeitsumgebungen virtueller Gemeinschaften häufig durch eine Steigerung des Sozialisierungsgrades und dem damit verbundenen Wunsch nach mehr Selbstadministration bzw. –regulation auf (vgl. [Hampel et al. 05]). Somit ist die Notwendigkeit zur Unterstützung dynamischer Kontexte zwar jeweils unterschiedlich begründet, führt aber bei der Umsetzung beider Systemklassen zu derselben konzeptionellen Herausforderung: Die technologische Infrastruktur muss möglichst reaktions- und anpassungsfähig bzgl. Kommunikation, Interaktion und Zusammenarbeit gestalten werden, um sowohl der individuellen als auch der kollektiven Entwicklungsdynamik ihrer Benutzer Rechnung zu tragen.

In einer interdisziplinären Kooperation zwischen den beiden Arbeitsgruppen Kooperative Medien und Kontextuelle Informatik und dem Decision Support & OR-Lab wurde an der Universität Paderborn eine modulare Architektur entwickelt, die sich als Basis für die Entwicklung solcher Lern- und Arbeitsumgebungen eignet.

Anstatt dabei ein konzeptionelles Datenbankschema für den speziellen Problembereich zu entwickeln und im Lebenszyklus der Anwendung weiter zu pflegen, wurde eine terminologiebasierte Komponentenentwicklung angestrebt: Die durchgehende Benutzung der von Hampel in [Hampel 01] konzipierten Metapher virtueller Wissensräume in spezialisierten Komponenten gestaltet diese einerseits sehr unabhängig voneinander, andererseits aber auch sehr kompatibel zueinander. Somit kann eine optimale Flexibilität und Erweiterbarkeit erreicht werden, um den Ansprüchen dynamischer und wachsender Plattformen zu genügen.

Der Beitrag beschreibt daher zunächst die konzeptionellen und technologischen Grundlagen, stellt unser Vorgehensmodell vor und gibt exemplarisch ein Beispiel für eine solche Komponente. Im letzten Teil ziehen wir ein Resümee und geben Ausblick auf weitere Entwicklungen.

## **2. Ein kollaboratives Objektmodell als Grundlage für die terminologiebasierte Komponentenentwicklung**

### **2.1 Die Metapher virtueller Wissensräume**

Räumliche Anordnung gehört zu den mächtigsten Interventionsfeldern der Wissensorganisation: Der Raum kanalisiert Kommunikation und ermöglicht oder beschränkt die Strukturierung, Verteilung und Vermittlung von Informationen (vgl. [Romhard 02]). Als zentrale Metapher bieten virtuelle Wissensräume (vgl. [Hampel 02]) Platz für das Arbeiten mit, sowie für die Ablage von Wissensobjekten. Sie können grundlegend frei und auf unterschiedliche Art und Weise verknüpft werden, wie z. B. durch Gänge, Türen oder durch Verweise zwischen Materialien, die in verschiedenen Räumen liegen. Da innerhalb eines Raums mittels Ordern oder Unterräumen weiterführende Strukturen möglich sind, können somit auch Mengen von Objekten, Personen und strukturelle Beziehungen semantisch, logisch und zeitlich abgebildet werden.

Das Konzept der Medienfunktionen identifiziert darauf aufbauend grundständige mediale Unterstützungsfunktionen, die technische Systeme zum Erzeugen, Verändern, Verknüpfen und Arrangieren von Wissensobjekten und ihren Strukturen bereitstellen.

### **2.2 Eine Objektmodell für die Wissensraummetapher**

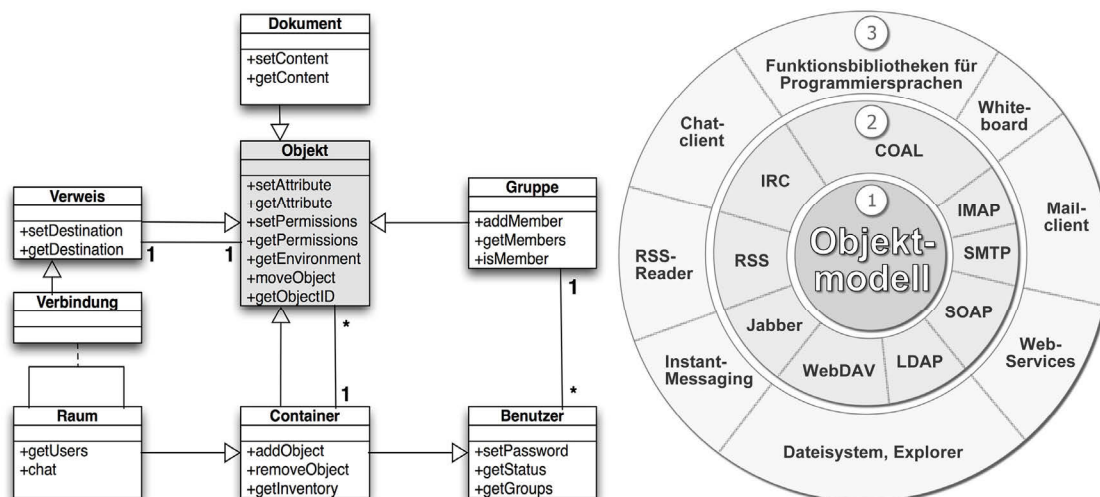
Aus Sicht der architektonisch-technischen Umsetzung der oben genannten Grundmetaphern hat sich in den letzten Jahren ein im Paderborner sTeam-System<sup>1</sup> umgesetztes Klassen- bzw. Objektmodell herauskristallisiert. Dieses beruht im Wesentlichen auf der Idee, kollaborative (CSCW-) Umgebungen als verknüpfte Raumstruktur zu verstehen, in der verschiedene Dokumente und geschachtelte Objektstrukturen abgelegt werden und sich Nutzer aufhalten und bewegen können.

Kennzeichnend für die Struktur des Kollaborationsservers ist das zugrunde liegende Objektmodell, welches sich auf die Objekttypen *Objekt*, *Raum*, *Verweis*, *Dokument*,

---

<sup>1</sup> sTeam ist ein Kollaborationsserver, der mit Unterstützung des Deutschen Forschungsnetzes am Heinz Nixdorf Institut konzipiert und als Open Source-Produkt umgesetzt wurde. Er ist fester Bestandteil der Debian Linux Distribution *Sarge*. Pakete für andere Unix-Systeme und weitere Informationen finden sich unter <http://www.open-team.org>

*Verbindung*, *Container*, *Benutzer* und *Gruppe* beschränkt, und eine Terminologie für Lern- und Arbeitsszenarien beschreibt (vgl. Abbildung 1).



**Abbildung 1: Das Objektmodell (links) stellt den generischen Kern (1) des sTeam-Kollaborationsservers dar. Medien- und Kommunikationsfunktionen werden darauf aufbauend über standardisierte Internet-Protokolle (2) technologisch offen gestaltet, was eine problemlose Integration in eine heterogene IT-Infrastruktur (3) und somit auch angepasste Sichten auf Räume und Wissensobjekte ermöglicht.**

Zentrales Element des Modells ist zunächst das *Objekt*. Es besitzt im Wesentlichen Funktionen, um Attribute zu setzen und auszulesen sowie Berechtigungen zu setzen und zu überprüfen. Alle weiteren Klassen erben die Grundeigenschaften von Objekten. Ein *Dokument* erweitert das Objekt um die Eigenschaft, einen Dokumentinhalt verwalten zu können.

*Verweise* ermöglichen es, Dokumente (bzw. Objekte) beliebig zu verknüpfen. Ein *Container* besitzt die Grundeigenschaft, Dokumente kapseln zu können. Im Wesentlichen werden Methoden bereitgestellt, um Objekte hinzuzufügen und den Inhalt aus Containern sowie seiner Umgebung zu erfragen.

Dergestalt ist der *Raum* eine Spezialisierung des Containers, in dem sich Nutzer aufhalten können. Räume können über *Verbindungen*, eine Spezialisierung von Verweisen, verknüpft werden. Auf diese Weise entsteht ein Korridor oder Verbindungsgang zwischen Räumen, in dem sich Nutzer entlang der gesetzten Berechtigung bewegen können.

Ein *Nutzerobjekt* erbt alle Eigenschaften des Grundtyps Objekt und eines Containers, ist also auf die Metapher eines Rucksacks übertragen in der Lage, beliebige Objekte mit

sich zu tragen und kann wie auch der Grundtyp *Objekt* mit beliebigen Attributen und Berechtigungen ausgestattet sein. *Gruppen* sind direkt von Objekten abgeleitet und erlauben es, verschiedene Teilnehmer einer Nutzergruppe zu verwalten. Über die Gruppenstruktur werden im Wesentlichen Berechtigungen gebündelt und Zugriffsstrukturen innerhalb von Kollaborationsumgebungen verwaltet. Über Zugriffsberechtigungen lassen sich verschiedene Interaktionen und Kontexte der Nutzung steuern, Nutzer bewegen sich in virtuellen Räumen und besitzen Berechtigungen an verschiedenen Objekten<sup>2</sup>. Hierüber lassen sich verschiedene Grade der Selbstorganisation von Gruppen und Nutzern realisieren (vgl. [Hampel 04]).

### **2.3 Integration durch angepasste Sichten**

Damit die vorgestellten Konzepte als grundlegend in einer heterogenen Umgebung dienen können, bezieht der Paderborner Kollaborationsserver sTeam die gängigen Kommunikations- und Infrastrukturprotokolle des Internets mit Hilfe von Protokolladaptern<sup>3</sup> auf die Elemente der Wissensraummetapher (vgl. Abbildung 1).

Verschiedene Clients können somit unterschiedliche, angepasste Sichten auf diese virtuelle Welt erzeugen. Auf diese Art können zum Beispiel Annotationsobjekte am Benutzer über Mail-Protokolle abgefragt, oder Dokumente in einem Raum mit Hilfe eines Shared Whiteboard-Clients in Gruppenarbeit gemeinsam umarrangiert und bearbeitet werden.

Insofern erzeugt auch die im nächsten Abschnitt skizzierte Komponentenarchitektur angepasste Sichten auf die Wissensraummetapher und die in der Persistenzschicht des Kollaborationsservers verwalteten Objektstrukturen.

## **3. Entwicklungsmodell terminologiebasierter CSCL/CSCW-Komponenten**

Die Implementierung verschiedener Lern- und Arbeitskontexte innerhalb eines Systems ist häufig mit einem hohen Aufwand und einer Vielzahl verschiedenster umzusetzender Funktionalitäten verbunden. Eine komponentenorientierte Architektur kann den

---

<sup>2</sup> In der späteren web-basierten Applikation äußert sich das darin, dass verschiedene Dokumente für den jeweiligen Nutzer sichtbar sind bzw. abgestufte Manipulationsmöglichkeiten an den jeweiligen Dokumenten existieren.

<sup>3</sup> Verschiedene Adapter übersetzen raumbasierte Nachrichten oder Aktionen in protokollgerechte Informationen (und vice versa). Ein Ereignissystem sorgt für die Weiterleitung von über ein Protokoll eingehenden Informationen an weitere Protokolladapter, so dass bspw. eine über einen IRC-Client geschickte Nachricht an die im Gruppenarbeitsraum anwesenden Mitglieder über das Shared Whiteboard oder mit dem Raum verbundene Jabber-Clients weitergeleitet werden kann.

---

Implementierungsaufwand wesentlich vereinfachen, indem verschiedene Softwaremodule zu einem Anwendungsszenario kombiniert bzw. aggregiert werden.

Komponenten besitzen – entgegen normalen Klassen – eine definierte Schnittstelle und müssen bestimmten Anforderungen bzgl. ihrer Unabhängigkeit, Abgeschlossenheit und Offenheit innerhalb dieser Architektur genügen (vgl. [Rautenstrauch 1999]).

Um eine möglichst große Kombinierbarkeit und somit auch Flexibilität zu gewährleisten, sollten die Komponenten technisch und fachlich kompatibel sein.

In diesem Abschnitt möchten wir unsere Erfahrungen in den Schritten zur terminologiebasierten Erstellung einer konfigurierbaren Softwarekomponente<sup>4</sup> auf Basis des verwendeten Kollaborationsservers sTeam näher beschreiben, bevor wir in Abschnitt 4.2 ein praktisches Beispiel einer solchen Komponente geben.

### **3.1 Organisatorisch-semantische Beschreibung des Szenarios**

Im ersten Schritt der terminologiebasierten Komponentenentwicklung liegt der relevante Unterschied zum herkömmlichen Vorgehen: Anstatt ein neues semantisches Datenmodell des durch die Komponente zu unterstützenden (Teil-)Problembereiches zu entwerfen, die grundsätzliche Datenstruktur dafür festzulegen und das so entstandene konzeptionelle Schema mit den bereits bestehenden Schemata der zu erweiternden CSCL- oder CSCW-Plattformen abzugleichen, wird das zu unterstützende Szenario mit der in Abschnitt 2.2 beschriebenen Terminologie konstruiert. Da somit das Objektmodell vorgegeben ist, entfällt auch die Festlegung von Objekten und Klassen sowie die Identifizierung von Strukturen, wie sie im Rahmen einer Objekt-orientierten Analyse normalerweise üblich wäre.

Als besondere Hilfe für die Modellierung des Konstruktes hat sich die Nutzung eines einfachen Diagrammschemas bewährt, das die Aufbau- und Ablauforganisation des Problembereichs mit Elementen der Wissensraummetapher abbilden kann (vgl. Abbildung 3).

Da das durch die Komponente zu verwaltende Konstrukt später zur Laufzeit eindeutig identifiziert und instantiiert werden muss, wird in diesem Schritt ein zentrales Objekt definiert. Über den eindeutigen Schlüssel dieser Objekte kann später auf das vollständige Konstrukt zugegriffen werden.

---

<sup>4</sup> Komponenten mit fachlichem Bezug werden auch Business „Objects“, „Anwendungselemente“ oder „Application Objects“ genannt (vgl. [Ortner 00], S. 2).

### 3.2 Implementierung des Konstruktes

Wie in Abbildung 1 dargestellt und in Abschnitt 2.3 beschrieben, sind die Instanzen der im Modell beschriebenen Objekte und ihre Funktionen über das Internet-Protokoll COAL zugänglich gestaltet. Hierauf bauen Schnittstellen auf, die die Terminologie der Wissensraummetapher und die dazugehörigen, in sTeam implementierten Funktionalitäten für verschiedene Programmier- und Skriptsprachen benutzbar machen. Auf Grundlage dieser Programmierschnittstelle (API) wird nun in der jeweiligen Sprache die Verwaltung des im ersten Schritt konzipierten Konstruktes umgesetzt. Hierzu stellt die sTeam-API Basisklassen in der Terminologie des virtuellen Wissensraums zur Verfügung. Dies gewährleistet, dass die Terminologie auch in den höheren Architekturschichten genutzt wird, was schlussendlich zu einer besseren Kompatibilität der Komponenten führt (vgl. hierzu Abschnitt 4.3).

In diesem Schritt werden nun auch die so genannten *Factory-Services* zum Erzeugen und Laden neuer Komponenteninstanzen definiert. Hierzu werden die von der Komponente verwalteten Konstrukte über den eindeutigen Schlüssel des zentralen Objektes aus der Persistenzschicht des sTeam-Kollaborationsservers ausgelesen und als Instanzen der Basisklassen in der Laufzeitumgebung zur Verfügung gestellt.

### 3.3 Funktionale Erweiterung der Komponente

In einem nächsten Schritt wird die Komponente sowohl um einfache Operationen wie get/set-Funktionen als auch um komplexere Funktionen erweitert, welche die fachliche Benutzung ermöglichen. Diese Funktionen bilden zusammen also die fachliche Schnittstelle der Komponente, das so genannte *Object-Interface*. Um eine größtmögliche Unabhängigkeit zu erzielen, sind die Datentypen dieser Schnittstellenart primitiv gehalten (String, Integer etc.); Instanzen von in sTeam persistent gespeicherten Objekten werden ausschließlich über die o.g. Basisklassen entgegengenommen bzw. zurückgegeben.

Wie schon in der Einleitung angedeutet, können verschiedene Lern- oder Gruppenarbeitsszenarien durchaus die gleichen fachlichen Funktionen benötigen, sich aber letztendlich durch den Freiheitsgrad an Selbstorganisation unterscheiden<sup>5</sup>. Soll eine Komponente in mehreren ähnlichen Szenarien zum Einsatz kommen, muss das in sTeam verwaltete Konstrukt um Attribute erweitert werden, um zusätzliche Informationen über Zustände, Steuerungsoptionen und Konfiguration persistent

---

<sup>5</sup> Anders ausgedrückt: Sie unterscheiden sich durch den Grad der vorgegebenen Strukturierung und Steuerung.

---

speichern zu können<sup>6</sup>. Hierzu sind get/set-Funktionen in der Komponente zu implementieren.

Ebenso wird in diesem Schritt eine weitere Schnittstellenart implementiert, nämlich die *Observer-Services*, welche auf eine von sTeam zur Verfügung gestellte Ereignissteuerung aufgesetzt werden können. Diese Services setzen Benachrichtigungsfunktionen zwischen Komponenten untereinander um.

Man beachte, dass die Kommunikation in web-basierten Anwendungen häufig über das zustandslose Protokoll HTTP erfolgt. Das bedeutet, dass in diesem Ausnahmefall eine ereignisorientierte Benachrichtigung über einen Browser als Client u.U. nicht erfolgen kann, da jede Verbindung zwischen Browser und Webserver eine eigenständige, unabhängige Transaktion darstellt. Eine Benachrichtigung an auf anderen Protokollen beruhenden Clients oder Anwendungen kann auf diesem Weg jedoch erfolgen (vgl. hierzu Abbildung 1).

### **3.4 Einbindung von Komponenten in Anwendungsszenarien**

Für die Umsetzung frei zu kombinierender Lernszenarien oder der auf die gewachsenen Anforderungen einer Community anzupassenden Arbeitsumgebungen werden die so erstellten Softwarekomponenten durch die Anwendungsfallsteuerung miteinander gekoppelt, wobei die Kommunikation zwischen Komponenten auch durch die Steuerung übernommen wird, um Abhängigkeiten untereinander zu vermeiden.

Eine mögliche persistente Speicherung gekoppelter Komponenten erfolgt ebenfalls in der vorgegebenen Terminologie; die von den Komponenten verwalteten Konstrukte werden in diesem Fall im virtuellen Wissensraum miteinander verbunden (vgl. Abschnitt 4.3).

## **4. Konfigurationsoptionen flexibler Komponenten am Beispiel des virtuellen Abgaberaums**

### **4.1 Computerunterstützung materialorientierter didaktischer Szenarien**

Materialien, die von Benutzern in einen Lern- oder Arbeitskontext eingebracht werden, können in verschiedenen Szenarien unterschiedliche didaktische Rollen spielen. Als Beispiele zählt Kleinen in [Pape et al., S. 162], Präsentationsmaterialien, Grundlagenmaterialien sowie Werkstücke und Arbeitsergebnisse auf.

---

<sup>6</sup> Wie in 2.2 beschrieben können durch sTeam Attribute an Objekten dynamisch verwaltet werden, ohne die Datenbankstruktur ändern zu müssen.



Die Anforderungen an die Computerunterstützung dieser materialorientierten Szenarien unterscheiden sich unter anderem bzgl. der gewünschten Interaktionsmöglichkeiten der Benutzer mit Materialien sowie den von der soziokulturellen Ausprägung des Szenarios abzuleitenden notwendigen Kommunikationsfunktionen.

In der Realität werden diese Szenarien jeweils durch eine physikalische Materialablage (z. B. Projektordner, Pinnwand, Abgabekasten für Übungszettel etc.) und einem Satz Regeln unterstützt, welche die Kommunikation, Interaktion und Zusammenarbeit darauf koordinieren sollen. Um diese einander ähnelnden Szenarien in verschiedenen Plattformen angemessen unterstützen zu können, wurde eine flexibel einsetzbare, konfigurierbare Komponente auf Basis der in Abschnitt 2.2 beschriebenen Terminologie entwickelt.

## 4.2 Eine Komponente für den virtuellen Abgaberaum

In der Metapher der virtuellen Wissensräume ausgedrückt lassen sich eine physikalische Materialablage und die zu ihrer Nutzung notwendigen Funktionen mit Hilfe eines Raums abbilden. Auf diesem Konstrukt aufbauend können koordinierende Regeln innerhalb der fachlichen Funktionen der Komponente und über zusätzliche Attribute des Raums verankert werden. Kommunikationsfunktionen sind in der vorgegebenen Terminologie durch Annotationen an Dokumenten möglich.

Neuen Abgaberaum anlegen

Bezeichnung:

Deadline: ja  nein  15 07 2005

Zugriffsvorbedingung

Zugriffsvorbedingung	Zustand vor der Deadline	Zustand nach der Deadline	
Eigenes Material abgegeben:	ja <input checked="" type="radio"/> nein <input type="radio"/>	ja <input type="radio"/> nein <input checked="" type="radio"/>	Workflow
Darstellung:	<input type="text" value="anonym"/>	<input type="text" value="Autor/en anzeigen"/>	Kommunikation
Annotationen:	<input type="text" value="anonym annotierbar"/>	<input type="text" value="annotierbar"/>	
Rechte für Eigenmaterial:	lesen <input checked="" type="checkbox"/> schreiben <input checked="" type="checkbox"/> löschen <input checked="" type="checkbox"/>	lesen <input checked="" type="checkbox"/> schreiben <input type="checkbox"/> löschen <input type="checkbox"/>	Interaktion
Rechte für Fremdmaterial:	lesen <input checked="" type="checkbox"/> schreiben <input type="checkbox"/> löschen <input type="checkbox"/>	lesen <input checked="" type="checkbox"/> schreiben <input type="checkbox"/> löschen <input type="checkbox"/>	

**Abbildung 2:** In der Anwendungsschicht des OpenSMT-Portals implementierter Konfigurationsdialog für die Komponente *Virtueller Abgaberaum*

Abbildung 2 zeigt den Konfigurationsdialog der im OpenSMT-Portal<sup>7</sup> eingebetteten Komponente, mit dessen Hilfe Dozierende die Nutzung dieses Softwarebausteins selbst bestimmen und somit die Lern- und Arbeitskontexte dynamisch erzeugen können. Wie man dem Bildschirmausschnitt entnehmen kann, hat der Dozierende die Möglichkeit, den Workflow, die Art und Darstellung der Kommunikation sowie die Interaktionsmöglichkeiten eines Benutzers mit Materialien zu definieren. Dabei haben sich die in Tabelle 1 beschriebenen Optionen in der Praxis als tauglich erwiesen, um über die Komponente viele Variationen von materialorientierten, didaktischen Szenarien abdecken zu können.

Option	Bedeutung
Deadline	Materialabgabe unterliegt einer zeitlichen Beschränkung. Ist eine solche definiert, kann die Komponente für den Zeitraum nach Ablauf des Termins abweichend konfiguriert werden.
Zugriffsvorbedingung	Definiert, ob Lernende vor dem Zugriff auf Material im virtuellen Raum selber Material dort abgelegt haben müssen.
Darstellung	Für den Fall, dass Lernende Materialien anderer Lernenden einsehen dürfen, legt diese Option fest, ob die Fremdmaterialien anonym oder mit Autor/Gruppe dargestellt werden.
Annotation	Schreibt vor, ob und wie Kommunikation zwischen Benutzern in Bezug auf die Materialien erfolgen kann: Anonym oder personifiziert
Rechte	Falls ein Zugriff auf Materialien durch den Lernenden möglich ist, definieren diese Rechte, wie er mit eigenem und fremdem Material interagieren kann: Lesen, Schreiben, Löschen <sup>8</sup>

**Tabelle 1: Konfigurationsoptionen des Virtuellen Abgaberaums**

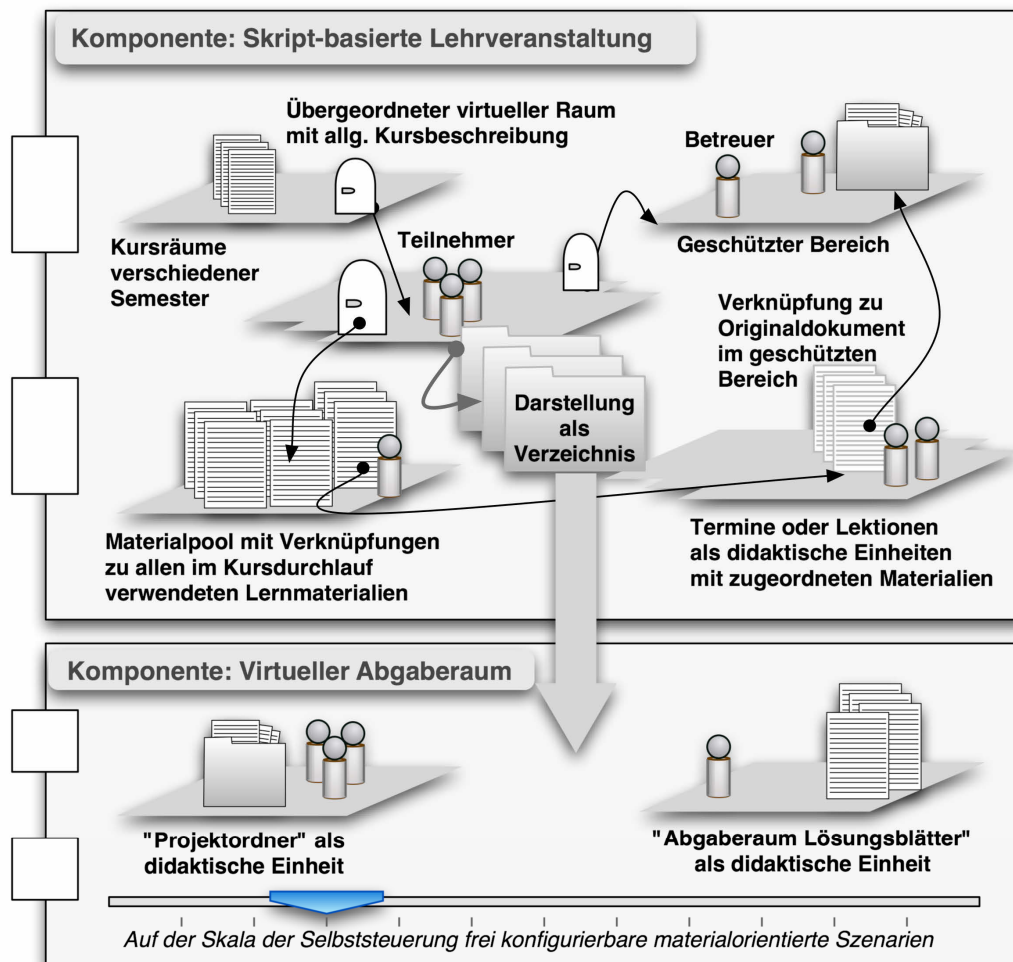
### 4.3 Einbettung des Abgaberaums in übergeordnete Szenarien

Um den Virtuellen Abgaberaum mit anderen didaktischen Szenarien kombinieren zu können, bedarf es einer losen Kopplung der entsprechenden Komponenten. Aus Gründen der geforderten Unabhängigkeit werden die Abhängigkeiten untereinander dabei sehr gering gehalten. Wie schon erwähnt, lassen sich terminologiebasiert entwickelte Komponenten in der Persistenzschicht über die dort verwalteten Konstrukte kombinieren. Eine Notwendigkeit, Komponenten in der Anwendungsschicht koppeln zu müssen, wird daher auf ein Minimum reduziert. Abbildung 3 zeigt ein Beispiel, bei dem mehrere Abgaberräume in ein Kursszenario eingebettet werden. Sie werden dabei als

<sup>7</sup> Portal des Rahmenwerks *Service-oriented Modules for Teachware* (OpenSMT). Für weitere Informationen siehe <http://www.opensmt.org>.

<sup>8</sup> Vgl. hierzu auch [Hampel 04].

Unterräume eines Kursraums angelegt; das zentrale Objekt des Konstruktes, auf dem die Komponente zur Kursdurchführung basiert. Die beiden beteiligten Komponenten benötigen keine weiteren Informationen voneinander, da die grundlegenden Basisklassen und ihre Funktionen ausreichen, um miteinander zu kooperieren.



**Abbildung 3: Terminologiebasiert entwickelte Komponenten lassen sich über die in der Persistenzschicht verwalteten Konstrukte kombinieren. In diesem Fall werden virtuelle Abgaberräume analog zu den Terminen und Lektionen als Unterräume des Kursraums angelegt.**

## 5. Praxiseinsatz und Ausblick

Lernumgebungen und Community-Plattformen müssen aus den in der Einleitung genannten Gründen häufig wechselnden und oft auch neuen Anforderungen genügen. Dies führt bei der Nutzung von konzeptionellen Datenschemata in Anwendungssystemen zu Problemen, denn diese sind im Anwendungssystembetrieb sowie in den

Ausbau der Anwendungen vollständig integriert und können nur mit sehr hohem Aufwand mit den entwickelten Anwendungen im ‚Einklang‘, also konsistent gehalten werden (vgl. [Ortner 00], S. 3).

Unsere Erfahrungen mit der skizzierten Architektur und der terminologiebasierten Komponentenentwicklung im Bereich der Lern- und Community-Plattformen haben gezeigt, dass diese allseits bekannten Nachteile hierbei nicht auftreten. Zum einen stieg die Produktivität bei den terminologiebasiert durchgeführten Systementwicklungen sogar stark an, zum anderen stellen wir immer wieder eine verbesserte Reaktionsfähigkeit auf dynamische Prozesse und die darauf folgenden wechselnden Anforderungen fest, was die Wartung dieser komplexen Systeme erheblich vereinfacht.

Die in diesem Beitrag skizzierte komponentenbasierte Architektur hat sich bereits in verschiedenen Kontexten bewährt, in denen virtuelle Lern- und Arbeitsgruppen systemübergreifend zusammen gearbeitet haben: Erstmals zum Einsatz gekommen ist sie zur Unterstützung des virtuellen Studienfachs VORMS<sup>9</sup>. In diesem Rahmen werden sowohl Präsenzkurse als auch rein virtuelle Kurse sowie Mischformen unterstützt. Die bislang mehr als 1500 Studierenden können sich zudem in universitätsübergreifenden Arbeits- und Lerngemeinschaften selbst organisieren (vgl. [Roth et al. 05]).

Des Weiteren wurde die Architektur in einem Kooperationsprojekt des Decision Support & OR-Labs und des Internationalen Instituts für Management der Universität Flensburg eingesetzt. Unter Begleitung der myconsult GmbH<sup>10</sup> wurde ein interdisziplinäres Forschungsprojekt durchgeführt, in dem die Nachfolgeplanung und -durchführung in klein- und mittelständischen Unternehmen untersucht, sowie ein IT-gestütztes Werkzeug zur Beratung erarbeitet wurde<sup>11</sup>.

Neben dem Einsatz in weiteren großen Community-Plattformen (vgl. z. B. den Beitrag „*Unterstützung selbst verwalteter Gruppenprozesse in virtuellen Gemeinschaften durch skalierbare Architekturkonzepte am Beispiel der Sifa-Community*“ in diesem Tagungsband) soll die Architektur auch im geplanten Infrastrukturausbau der Universität Paderborn<sup>12</sup> genutzt werden. Im Fokus des Projektes mit dem Namen Locomotion (*Low-Cost Multimedia Organisation and Production*) steht dabei neben der Prozessunterstützung der *Modul- und Prüfungsverwaltung* die verstärkte Nutzung von

---

<sup>9</sup> Siehe <http://www.vorms.org>

<sup>10</sup> siehe <http://www.myconsult-team.de>

<sup>11</sup> Die Arbeit wurde in Form eines Distance Learning Projektes durchgeführt, bei dem 20 angehende Betriebswirte, Wirtschaftsinformatiker und Informatiker beider Universitäten neben Präsenzphasen modernste Kommunikationsmittel wie Videokonferenzsysteme und virtuelle Lernumgebungen nutzen konnten.

<sup>12</sup> Ein durch das BMBF im Förderprogramm *eLearning-Dienste für die Wissenschaft* unterstütztes Projekt mit einer Laufzeit von 3 Jahren. Für weitere Informationen siehe <http://locomotion.uni-paderborn.de>

*eLearning, eTeaching* und *eCollaboration*, um die Qualität der Lehre und des Lernens zu steigern sowie die Abwicklung der damit verbundenen Prozesse zu optimieren.

## Literatur

- Ortner, E. (2000). *Terminologiebasierte, komponentenorientierte Entwicklung von Anwendungssystemen*. In: Flatscher, R.G., Turowski, K. (Hrsg.): Tagungsband 2. Workshop komponentenorientierter betrieblicher Anwendungssysteme, Wien, 02/2000, S. 1-20
- Hampel, T. (2002). *Virtuelle Wissensräume – Ein Ansatz für die kooperative Wissensorganisation*. Universität Paderborn, Fachbereich 17 – Informatik, Dissertation, Paderborn
- Hampel, T. (2004). *Access Rights – The Keys to Cooperative Work/Learning*. In: Hicks, D. L. (Hrsg.): *Metainformatics. International Symposium, MIS*. Springer Verlag, Salzburg, S. 14-31
- Hampel, T., Kahnwald, N., Roth, A., & Köhler, T. (2005). *An Adaptable Platform for Evolving Communities of Practice*. In: Tagungsband des Workshops: "Design for Large-Scale Digital Communities", 2nd International Conference on Communities and Technologies, Milano, Italien, 06/2005
- Pape, B., Krause, D., & Oberquelle, H. (2004). *Wissensprojekte – Gemeinschaftliches Lernen aus didaktischer, softwaretechnischer und organisatorischer Sicht*. Waxmann, Münster
- Rautenstrauch, C., et al. (1999). *Fachkomponenten zur Gestaltung betrieblicher Anwendungssysteme*. In: *Informationsmanagement & Consulting* 14/2, S. 25-34
- Romhardt, K. (2002). *Wissensgemeinschaften – Orte lebendigen Wissensmanagements*. Versus Verlag, Zürich
- Roth, A., Hampel, T., & Suhl, L. (2005). *Von serverzentrierten Lernobjekten zu kooperativen Wissensobjekten*. In: Tagungsband des 3. Workshops GML 2005, Grundfragen Multimedialer Lehre, Cottbus, 03/2005 (im Druck)
- Roth, A., & Suhl, L. (2005). *Plattformübergreifende Architekturen in föderativen E-Learning-Umgebungen*. In: Breitner, M. (Hrsg.): *E-Learning – Einsatzkonzepte und Geschäftsmodelle*. Springer Verlag, Berlin, Heidelberg