Softwarearchitektur für den ViKar-Campus

Dipl.-Inform. Sven Claußen

Zentrum für Multimedia
Fakultät für Informatik
Universität Karlsruhe

Karlsruhe, 15.2.2001

Inhalt

Autor:

Dipl.-Inform. Sven Claußen
Universität Karlsruhe
Fakultät für Informatik
Email:claussen@ira.uka.de
http://zemm.ira.uka.de

I. Übersicht

In Karlsruhe wurde 1998 das Projekt „Virtueller Hochschulverbund Karlsruhe“ [ViKar] ins Leben gerufen, in dem die 6 ortsansässigen Hochschulen (Universität Karlsruhe, Hochschule für Musik, Fachhochschule, Berufsakademie, Hochschule für Gestaltung/ZKM und die Pädagogische Hochschule Karlsruhe) den Aufbau einer gemeinsamen, virtuellen Hochschule – die keine juristisch eigenständige ist - in Karlsruhe vorantreiben. Hierbei soll in der ersten Phase die Qualität des Präsenzstudiums für alle Partner durch elektronische, multimediale Mehrwertdienste erhöht werden. Im weiteren Ausbau wird auch ein teleorientiertes Studium im ViKar möglich sein.
Das Projekt ViKar, das durch das Land Baden-Württemberg gefördert wird, ist in der Durchführung in mehrere Teilprojekte unterteilt worden. Dabei beschäftigt sich ein Teilprojekt beispielsweise mit der Evaluation der Inhalte, ein anderes bereitet bestehende Lehrmaterialien multimedial auf. Das Teilprojekt 1 hat den Aufbau der technischen Infrastruktur zum Ziel und wird von der Universität Karlsruhe bearbeitet. Innerhalb dieses Teilprojekts soll die Lernumgebung entstehen, die sowohl die Lehrenden bei der Erstellung von Kursmaterialien unterstützen als auch die Lernenden durch die Kurse und Lehrveranstaltungen führen soll.
Die ViKar-Lernumgebung ist hierbei die Summe aller virtuellen Einrichtungen und Dienste für das virtuelle Studium. Virtuelle Einrichtungen sind beispielsweise die elektronische Bibliothek, der Infokiosk, der Lernserver, die Hochschulverwaltung, der Studienarbeitsplatz und das virtuelle Labor. Virtuelle Dienste unterstützen die Kommunikation, dienen zur Benutzung der virtuellen Einrichtungen und zur Unterstützung des Studiums. Auch externe virtuelle Dienste können eingebunden werden. Das Online-Studium findet im Wesentlichen am Studienarbeitsplatz statt, von dem aus Dienste und Inhalte in Anspruch genommen werden. Der Lernserver dient als Dokumentenverwalter und Dienstegeber. Ein elektronischer Studienassistent fungiert als Begleiter und Dienstleister für den Studierenden.
Im Folgenden soll, ausgehend von den angebotenen Diensten des ersten ViKar-Prototyps, die verwendete Softwarearchitektur für den ViKar-Campus aufgezeigt und kritisch untersucht werden. Ein abschließender Ausblick gibt Empfehlungen für den weiteren Ausbau des ViKar-Campus.

II. Entwicklungsphasen

Die Überlegungen für den ViKar-Campus gingen ursprünglich von einem realen Campus aus. Die dort vorhandenen Dienste und Einrichtungen - für die Studierenden reichlich vertraut - lassen sich vielfach auf virtuelle Komponenten projezieren. So wird beispielsweise auch ein virtueller Campus eine Verwaltung beinhalten, wo sich die Studierenden einmalig registrieren lassen müssen. In einem virtuellen Labor können Studierende Versuche und Simulationen durchführen und so angelerntes Wissen direkt am Objekt überprüfen.
Auf der anderen Seite beinhaltet ein virtueller Campus auch die Chance, Komponenten aufzunehmen, die im realen Leben so gar nicht denkbar sind. Einem elektronischen Studienassistenten könnten z.B. Aufgaben übertragen werden, die zwar nötig sind aber im realen Leben auch viel Zeit kosten würden. Dazu können eine Anmeldung zu Prüfungen, Rechercheaufgaben und eine Vermittlung von Experten zu einem bestimmten Wissensgebiet gehören.

a) "Naive" Phase (1995 - 1997)

In der "naiven" Phase wurden an der Fakultät für Informatik der Universität Karlsruhe erste Erfahrungen mit einem Lernserver (Abb.1) gesammelt, der für alle Lehrveranstaltungen des Informatik-Studiums nach dem "Schubladen-Prinzip" Verweise, Literaturhinweise, Übungsblätter, Volltextdokumente, PowerPoint-Folien u.v.m. beinhaltete. Technisch stand in dieser Phase noch ein einfacher Webserver (Apache) im Mittelpunkt, der um einige CGI-Scripte ergänzt und so erste sinnvolle Dienste erbringen konnte. Ein einfaches Blackboard diente bereits als Kommunikationskomponente.

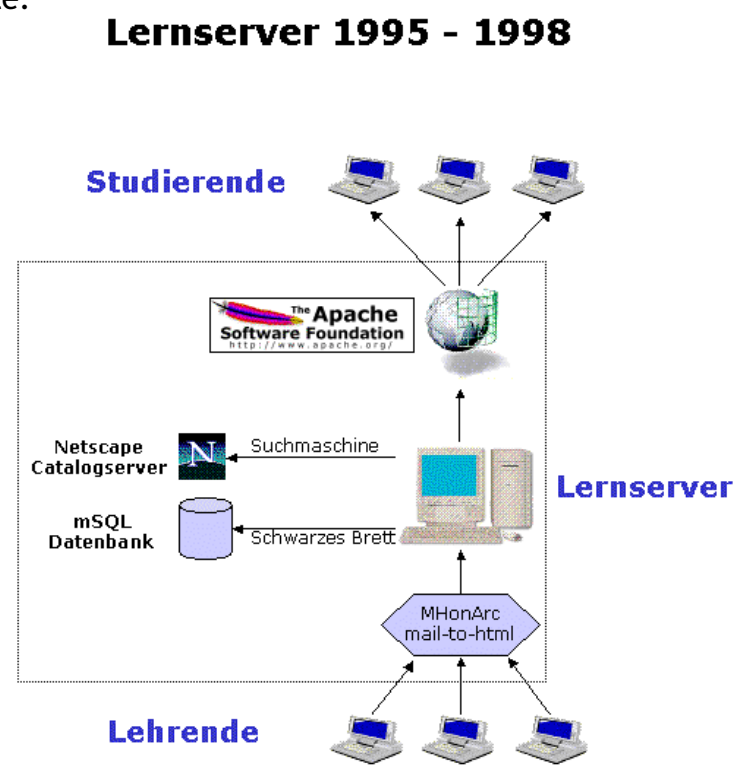

Abb.1: Lernserver (1995-1998)

Abb.2: "Naives" Bild (1996)

Parallel hierzu wurde bereits die Projektion realer Campus-Komponenten auf einen virtuellen Campus vorgenommen und in einem "naiven" Bild (Abb.2) festgehalten (von diesem wurde der Name dieser Entwicklungsphase abgeleitet). Hierbei wurden 9 Basiskomponenten definiert:

- Elektronische Bibliothek
- Lernserver
- Multimedia-Produktion
- Infokiosk
- Kommunikation
- Studienarbeitsplatz
- Virtuelles Labor
- Anmeldung
- Hochschulverwaltung

In Tabelle 1 werden diese Komponenten nochmals einzeln kurz vorgestellt und stichwortartig erläutert.
Das "naive" Bild hat sich trotz mancher Unzulänglichkeiten bis heute als ein gutes Leitbild für das ViKar-Projekt erhalten.

Tabelle 1: Basiskomponenten des Virtuellen Campus´:

	**Elektronische Bibliothek**	Zugriff auf Volltextdokumente, Videos, Literaturverweise, -recherche, reale/virtuelle Bibliothekskonten bis hin zur intelligenten Suchmaschine
	**Lernserver**	Beinhaltet Lehr- und Lernmaterialien aller angebotenen Lehrveranstaltungen, Wissensüberprüfungen, Glossar und Schnittstellen zu externen Systemen Leistet die Strukturierung der angebotenen Kurse
	**Multimedia-Produktion**	Dient der Produktion von Lehr- und Lernmaterialien, beinhaltet eine Multimedia- und eine KnowHow-Datenbank Leistet Formatkonvertierungen und Qualitätsprüfungen Einbindung von Autorenwerkzeugen
	**Infokiosk**	Bietet diverse Schwarze Bretter (z.B. Lernpartnervermittlung, Austausch von Lernmaterialien, Kulturkalender, Campus-Mitteilungen etc.)
	**Kommunikation**	Drehscheibe für die Kommunikation zwischen Lernenden - Lehrenden, unterstützt synchrone und asynchrone Kommunikationsmöglichkeiten
	**Studienarbeitsplatz**	Eigentlicher Lernarbeitsplatz des Studierenden, personalisierter Zugriff auf "gebuchte" Lehrveranstaltungen, Ablage von individuellen Lernmateriellen (Bookmarks, Übungsblätter etc.) Beinhaltet Persönlichen Studienassistenten
	**Virtuelles Labor**	Dient der Bereitstellung und Vermittlung von virtuellen Experimenten und Simulationen
	**Anmeldung**	Anmeldung und Authentifizierung des Benutzers am Campus Leistet auch Sitzungsverwaltung
	**Hochschulverwaltung**	Dient zu Administrationszwecken (z.B. Abrechnung) sowie der Verwaltung der Benutzer (inkl.-rollen) und Ressourcen

b) Evaluationsphase (1997 - 1999)

Nach ersten positiven Erfahrungen mit einem Lernserver (insbesondere seitens der Studierenden) wurde der Lernserver auf ein Dokumentenmanagementsystem aufgesetzt. Dabei wurden seinerzeit zahlreiche Lernsysteme auf eine Eignung hin untersucht. Keines konnte allerdings mit einem vollen Funktionsumfang überzeugen.
Am geeignetsten erschien bei der damaligen Evaluation das Hyperwave Training Space [HTS] als Aufsatz auf das Dokumentenmanagementsystem Hyperwave Information Server [HWIS]. Dabei spielte auch die großzügige Lizenzpolitik der Firma Hyperwave gegenüber Hochschulen eine positive Rolle. Der Lernserver der Fakultät für Informatik der Universität Karlsruhe wurde daraufhin auf einen Hyperwave Information Server umgestellt. Da sich Hyperwave Training Space noch mitten in der Entwicklung befand, wurde HTS zunächst für den realen Betrieb noch nicht eingesetzt.
Die Einbindung einer Suchmaschine (htDig) und des internen Verwaltungssystems der Fakultät (i3V) sowie die Nutzung der integrierten Funktionalitäten des HWIS (z.B. Benutzerverwaltung, Schwarze Bretter, automatische Verweiskontrolle, Trennung von Inhalt und Meta-Daten etc.) brachten bereits wesentliche Verbesserungen für Studierende und Lehrende mit sich.

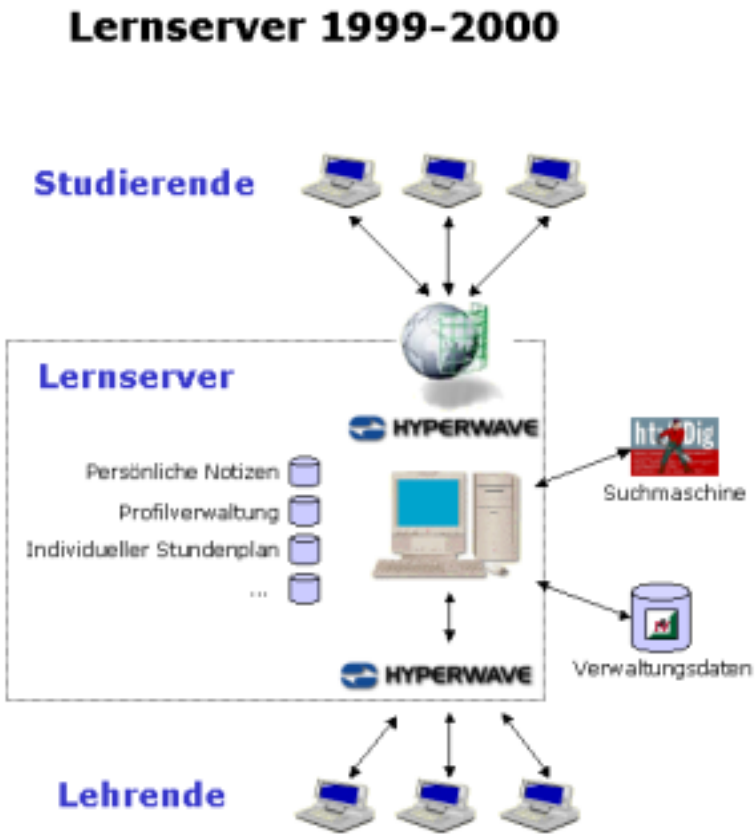

Abb.3: Lernserver (1999-2000)

Die guten Erfahrungen mit einem kommerziellen System (das dazu kostenfrei für Hochschulen ist) und die positiven Ansätze, die Hyperwave Training Space (bzw. Gentle-WBT, wie es später genannt wurde) zeigte, half schließlich bei der Entscheidung, für die Entwicklung des 1.ViKar-Campus-Prototypen Gentle-WBT als Basissystem des Virtuellen Studienarbeitsplatzes einzusetzen.

III. ViKar-Campus

Der erste Prototyp eines ViKar-Campus´ (kurz: 1.ViKar-Prototyp) wurde im Jahr 2000 entwickelt und sollte einen ersten Eindruck von dem Potenzial vermitteln, das im ViKar-Campus steckt bzw. stecken soll. Dabei stand auch hier das eingangs vorgestellte "naive" Bild als Leitbild für die einzelnen Komponenten "Pate". Daneben sollte der 1.Prototyp allerdings auch eine Sitzungsverwaltung und eine erste Version des Persönlichen Studienassistenten beinhalten.

a) Technologie

Als Basistechnologie wurde PHP4 gewählt. Die Benutzerschnittstelle wird durch einen kostenlos erhältlichen Webserver (Apache) mit eincompiliertem PHP4-Modul gebildet. Mit PHP liegt eine sehr leistungsfähige Scriptsprache vor, die schnell zu erlernen ist, kostenlos erhältlich ist und die wichtigsten Vorteile aller verfügbaren Script- und Programmiersprachen in sich vereinigt. Der Code wird direkt auf dem Serverrechner vom Webserver (Apache) interpretiert. An den Client wird HTML-Code (evtl. inkl. Javascript) geliefert. PHP besticht außerdem durch die Unterstützung einer Vielzahl von gängigen Datenbanken (u.a. Oracle, LDAP, Hyperwave, MySQL) und aufgrund der großen Verbreitung durch ein reichhaltiges Angebot von OpenSource-Anwendungen. Die gesamte Sitzungsverwaltung konnte auf einfache Art und Weise mittels PHP realisiert werden.
Neben der Sitzungsverwaltung werden folgende Komponenten des "naiven" Bildes durch PHP realisiert:

Anmeldung	Anmeldung des Nutzers durch ein einfaches HTML-Formular, Überprüfen der eingegebenen Werte in der LDAP-Datenbank, anschließend Zuweisung der Nutzerrolle, Einrichten einer Sitzung mit globalen Sitzungsvariablen
Hochschulverwaltung	Administration der Accounts (Neueinrichtung, Bearbeitung, Löschung), Zuweisen der Nutzerrollen
Infokiosk	Einfaches Schwarzes Brett, realisiert mit Unterstützung einer MySQL-Datenbank, Verknüpfung mit Kommunikationskomponente
Lernserver	Direkter Zugriff auf Lerninhalte des Hyperwave Information Servers über die PHP-Schnittstelle (Studierende) bzw. Hyperwave-Schnittstelle(n) (Lehrende)

Das als "Herzstück" ausgewählte Lernsystem ***Gentle-WBT*** der Uni Graz (seit Herbst 2000 im Vertrieb der Firma Hyperwave als ***eLearning Suite***) deckt im wesentlichen die Funktionalität des Virtuellen Studienarbeitsplatzes ab. Ein Zugriff auf die Lernmaterialien ist so direkt über den Virtuellen Studienarbeitsplatz möglich. Auch die Kommunikationskomponente ist bereits im Gentle-WBT integriert. Übungen (insbesondere Simulationen und Online-Versuche) können ebenfalls mittels Hyperwave/Gentle-WBT angeboten werden.
Der Persönliche Studienassistent baut auf einer Agentenarchitektur der Universität Ulm (CIA) auf und wird komplett in Java entwickelt.
Als Datenbanken werden neben dem von Gentle-WBT benötigten Hyperwave Information Server eine MySQL-Datenbank zur schnellen und einfachen Zwischenspeicherung von temporären Nutzerdaten sowie eine LDAP-Datenbank für die Benutzerverwaltung eingesetzt.
Weitere (externe/interne) Datenbanken können aufgrund standardisierter Schnittstellen (i.d.R. SQL) beliebig hinzugenommen werden.
Das allem zugrunde liegende Betriebssystem ist Linux.

b) Architektur

Die der Softwarearchitektur des ViKar-Prototypen zugrunde liegenden Komponenten müssen folgenden Anforderungen im Wesentlichen genügen:

- Kostengünstig/-frei (finanzieller Aspekt)
- Vorhandene Standard-Schnittstellen (Erweiterbarkeit)
- Skalierbar ("Mitwachsen")
- Anpassbar (Integrierbarkeit)

Unter diesen Gesichtspunkten soll die bereits oben in groben Zügen vorgestellte Softwarearchitektur nochmals näher vorgestellt werden (vgl. Abb.4):

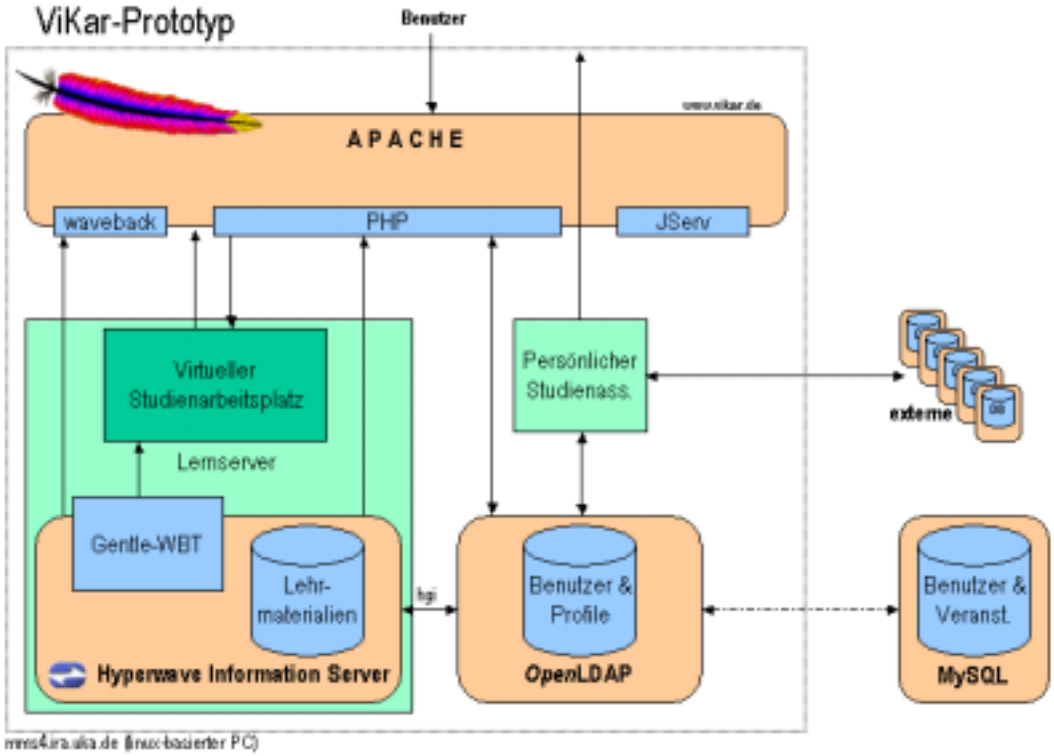

Abb.4: Serverarchitektur

Komponente	Kosten	Schnittstellen	Skalierbarkeit	Anpassbarkeit
Apache	kostenfrei	u.a. CGI, Schnittstellen durch Module (z.B. PHP, JServ)	Lastverteilung auf mehrere Rechner	Eincompilierung von Modulen
PHP4	kostenfrei	Scriptsprache, als Apache-Modul erhältlich, Unterstützung aller gängigen Datenbanksysteme	als Modul ist PHP direkter Bestandteil von Apache	eigene PHP-Funktionen
HWIS	für Hochschulen kostenfrei	Waveback, Oracle, PHP-Bibliothek, LDAP-Modul	Lastverteilung auf mehrere Rechner	Editieren von Templates, API
Gentle-WBT	für Hochschulen kostenfrei	s. HWIS	---	Editieren von Templates & Code
OpenLDAP	kostenfrei	LDAP, PHP-Bibliothek, Java	Lastverteilung auf mehrere Rechner	bel. Verzeichnisstruktur möglich
MySQL	kostenfrei	SQL, PHP-Bibliothek, Java	nur eingeschränkt	bel. Datenstruktur möglich, eigene Funktionen, API
CIA	für Projektpartner kostenfrei	durch eigene Agenten	Lastverteilung auf mehrere Rechner	eigene Agenten hinzufügen

IV. Ausblick

a) Beurteilung

Aufgrund der gewonnenen Erfahrungen können für die Weiterführung des Projekts folgende Schlüsse gezogen werden:
Die Verwendung von PHP4 hat sich als sehr erfolgreich gezeigt. Grundsätzlich sollten die Systemanforderungen weg von den Clients und den Netzverbindungen hin zu den Serverkomponenten verlagert werden. Dies wird durch ein serverseitiges Ausführen von Programmcode weitestgehend erreicht. Dabei wird an den Client letztlich nur HTML-Code (höchstens inkl. Javascript) und - sofern es sich nicht gänzlich vermeiden lässt - Java-Applets mit einfachstem, schlanken Java-Code (das möglichst auch auf Browsern 3.x lauffähig ist) übersandt. Statt HTML ist selbstverständlich mittelfristig XML in jedem Fall zu bevorzugen (durch die noch eingeschränkte Funktionalität heutiger Webbrowser kurzfristig jedoch nur serverseitig einsetzbar).
Die Verwendung von schlüsselfertigen Campuskomponenten erspart eine Menge von Entwicklerarbeit, verursacht jedoch einen teilweise recht hohen Integrierungsaufwand, um ein einheitliches "Look & Feel" zu erreichen. Dies ist gegeneinander aufzuwiegen. Im Falle von Gentle-WBT ist mittelfristig eine Ersatzlösung zu suchen, bzw. selber zu entwickeln. Durch die sehr aufwendige Anpassung von Gentle-WBT, bei der teilweise sogar Teile des Codes verändert werden mussten, ist man an eine bestimmte Version gebunden oder muss den Anpassungsaufwand stets wiederholen.
Die sehr schwankende Lizenzpolitik (von uneingeschränkt kostenpflichtig [Stand: Anfang 2000] bis komplett kostenfrei [Stand: Ende 2000]) der Firma Hyperwave zeigt zudem die Nachteile einer technologischen (und finanziellen) Abhängigkeit deutlich auf. Anzustreben sind daher Eigenentwicklungen (allerdings mit dem Nachteil eines hohen Implementierungsaufwandes) bzw. kostenfreie OpenSource-Komponenten.
Auf der anderen Seite kann eine direkte Partnerschaft mit einem kommerziellen Unternehmen durchaus von Vorteil sein, da auf diesem Wege die weitere Entwicklung des Produkts langjährig gesichert ist. Entwicklungsaufwand kann so von innen nach außen verlagert werden. Letztlich entscheidet jedoch auch die verwendete Technologie über den Erfolg eines Produktes und damit letztlich über die sinnvolle Verwendung im ViKar-Projekt.
Um kein abgeschlossenes System zu erhalten, dessen Anpassung an veränderte Außenbedingungen bzw. an integrierte Neukomponenten nur äußerst aufwendig zu realisieren ist, ist schließlich unbedingt auf eine modulare und offene Entwicklung zu achten. Es müssen hierbei möglichst standardisierte Schnittstellen (sowohl interne als auch externe) vorhanden sein.
Unterschiedliche Kategorien von Daten (Lehr-/Lerndaten, beschreibende Metadaten, Benutzer-/Verwaltungsdaten) sind in möglichst einfachen (standardisierten) Datenmodellen voneinander zu trennen und bevorzugt datenbankbasiert zu halten. So wird auf Kursebene eine Integration von extern entwickelten (und ebenfalls standardisierten) Lernkursen bzw. -modulen erst realisierbar. Eine sinnvolle Autorenumgebung ist hierbei natürlich ebenfalls von erheblichem Vorteil und in der angestrebten Art im ersten ViKar-Prototyp noch nicht vorhanden.

b) Weiterentwicklung

In der mittelfristigen Weiterentwicklung ist der Virtuelle Campus von einem Prototypen zu einer alltagstauglichen Lernumgebung zu führen. Auf dem Weg dorthin wird der 2.Prototyp, der unter Berücksichtigung der bislang gewonnenen Erfahrungen bei der Entwicklung des 1.Prototypen zu realisieren ist, richtungsweisend sein. Bislang verwendete Technologien sind einer erneuten Prüfung zu unterziehen und mit inzwischen weiter entwickelten Technologien zu vergleichen. Insbesondere die Integration externer kommerzieller Systeme ist hinsichtlich des Integrierungsaufwandes auf der einen Seite und des einzusparenden Entwicklungsaufwandes auf der anderen Seite zu prüfen.
Da eine modulare Entwicklungsstrategie in jedem Fall einzuhalten ist, wird es zukünftig auch verstärkt auf die Verwendung von standardisierten Schnittstellen ankommen. Möglichst alle Daten sind in standardisierten Datenbanken zu halten. LDAP und MySQL unterstützen diese Forderung bereits. Lern-, Nutzer- und Metadaten sind voneinander zu trennen. Daher sollte verstärkt auf XML gesetzt werden. Außerdem sollte LDAP die alleinige Datenbank für Nutzerdaten sein.
Die Komponenten sind auch hinsichtlich ihrer Funktionalität zu überprüfen und zu bewerten. Die Kommunikationskomponente, die im 1.Prototyp im wesentlichen im Studienarbeitsplatz integriert ist (Gentle-WBT), ist herauszulösen und verstärkt auszubauen, da hier die Grundlage für eine breite Akzeptanz des ViKar-Campus durch die Nutzer vermutet wird (neben hochwertiger Lerninhalte). Verbale Kommunikation ist dem einfachen Chat in jedem Fall hinzuzufügen. Whiteboardfunktionen sind ebenso wichtig wie beispielsweise ein Instant Messaging System, um die bilaterale Kommunikation weiter zu fördern.
Grundsätzlich sind anforderungsintensive Anwendungen verstärkt vom Client zum Server zu verlagern. Dem Client sind möglichst nur reine HTML-Seiten (inkl. Javascript) zuzumuten. PHP folgt diesem Konzept bereits zufriedenstellend. Auch Java Server Pages / Servlets genügen dieser Anforderung und kommen daher als Technologie durchaus in Betracht. Serverseitiges Javascript ist dagegen aufgrund langsamer Interpretergeschwindigkeit und der eingeschränkten Unterstützung von Basistechnologien eher abzulehnen.
Schließlich war auch der Einsatz von CIA als Basistechnologie für den Studienassistenten sehr problematisch und es bleibt zu prüfen, ob der bereits mit wenigen Campusnutzern überforderte Server aufgrund der großen Anzahl von zu startenden Java-Prozessen entlastet werden kann. Eventuell ist der Persönliche Studienassistent durch ein "javafreies" System zu ersetzen bzw. ebenfalls durch Java Server Pages / Servlets zu realisieren. Dadurch würden sich auch die Systemanforderungen an den Clientrechner auf ein Minimum reduzieren.

V. Literatur

a) Lernserver

[Clau97] S. Claußen, P. H. Schmitt
„Education via Nets"
MeDoc-Klausurtagung in Dagstuhl, Juli 1997

[Clau99-1] S. Claußen, A. Wolf, D. Saqe
„Kurzanleitung Lern-Server v.2.0"
Universität Karlsruhe, April 1999

[Clau99-2] S. Claußen
„Lern-Server und elektronischer Studienassistent als virtuelle Dienstleister in der ViKar-Lernumgebung"
Universität Karlsruhe, Juli 1999

[Die98a] Th. Dietinger, H. Maurer
„GENTLE – General Network Training an Learning Environment"
Proc. of EDMEDIA98/ED-TELECOM98, S.274 – 280, Freiburg, Juni 1998
http://wbt.iicm.edu/gentle/papers/edmedia98.pdf

[Die98b] Th. Dietinger, H. Maurer, M. Pivec
„Multimedia Learning Environment: Combining easier courseware production and new learning methods"
Proceedings of XV. IFIP World Computer Congress, Aug/Sep 1998
http://wbt.iicm.edu/gentle/papers/ifip98.pdf

[Die98c] Th. Dietinger, Ch. Gütl, H. Maurer, M. Pivec, K. Schmaranz
„Intelligent Knowledge Gathering and Management as New Ways of an Improved Learning Process"
Proc. of WebNet98, Orlando/Florida, Nov. 1998
http://wbt.iicm.edu/gentle/papers/webnet98.pdf

[Die98d] Th. Dietinger, Ch. Gütl, H. Maurer, M. Pivec
„GENTLE – Die sanfte Einführung in virtuelle Ausbildung"
Proc. of ICL98, Villach, S.11-17, 1998

[End95] M.Enderle, G.Ferch
„Entwicklung und Aufbau eines WWW-Servers zur Unterstützung des Lehrbetriebs der Fakultät für Informatik"
Teamstudienarbeit, Karlsruhe 1995

b) Technologie

[Boger99] M. Boger
„Java in verteilten Systemen"
dpunkt.verlag, Heidelberg 1999, ISBN 3-932588-32-0

[Dicken00] H. Dicken, G. Hipper, P. Müßig-Trapp
„Datenbanken unter Linux"
MITP-Verlag, Bonn 2000, ISBN 3-8266-0555-1

[Green98] P. Greenspun
„Datenbankgestützte Web-Sites"
Hanser-Verlag, München 1998, ISBN 3-446-19341-3

[Krause00] J. Krause
„PHP - Grundlagen und Lösungen"
Hanser-Verlag, München 2000, ISBN 3-446-21301-5

[Midd96] S. Middendorf, R. Singer, S. Strobel
„JAVA: Programmierhandbuch und Referenz"
dpunkt.verlag, Heidelberg 1996, ISBN 3-920993-38-1

[Schl99] J. Schlierf, R. Weber
„Programmieren mit Swing"
Hanser-Verlag, München 1999, ISBN 3-446-21151-9

c) CIA

[Kargl99-1] Kargl, Illmann, Weber
„CIA - a Collaboration and Coordination Infrastructure for Personal Agents"
Proceedings of the IFIP TC6 WG6.1 Second International Working Conference on Distributed Applications and Interoperable Systems (DAIS'99), Juni-July 1999, Helsinki, Finland

[Kargl99-2] Kargl, Illmann, Weber
„Evaluation of Java Messaging Middleware as a Platform for Software Agent Communication"
Java Informations Tage (JIT'99), September 1999, Düsseldorf, Germany

[Kargl99-3] Kargl, Illmann, Weber, Ribhegge
„Dynamic User Interfaces with Java"
Proceedings of the Webnet 99, October 1999, Honolulu, Hawaii

[Illm99] Illmann, Kargl, Weber
„Design of an Agent Cluster as Integrative Environment of Personal Agents"
Proceedings of the ICIIS 99, November 1999, Washington, USA

[Schm98-1] Schmidt, Specker, Partsch, Weber, Höck
„An Agent-based Telecooperation Framework"
Proceedings of the First International Workshop on Cooperative Buildings, CoBuild'98, February 1998

[Schm98-2] Schmidt, Specker, Partsch, Weber
„Agents, Brokers, Traders, and Services in Cooperative Systems"
Proceedings of the Third International Conference on the Design of Cooperative Systems, COOP'98, May 1998