

# CASA - Ein Konzept zur nutzergetriebenen, kontextbasierten Dienstintegration in Lehr- und Lernumgebungen

Philipp Lehsten, Djamshid Tavangarian

Forschungsgruppe Rechnerarchitektur  
Institut für Informatik, Universität Rostock  
Joachim-Jungius-Str. 9  
18059 Rostock  
philipp.lehsten@uni-rostock.de  
djamshid.tavangarian@uni-rostock.de

**Abstract:** An Hochschulen ist heute eine hochgradig heterogene IT-Landschaft mit Diensten und Web-Anwendungen etabliert. Durch die gewachsenen Strukturen an vielen Fakultäten und die häufig dezentral verwalteten Anwendungen ergibt sich für die Studierenden die Notwendigkeit, eine Vielzahl von Diensten zu kennen und regelmäßig aufzusuchen. Das in diesem Beitrag vorgestellte CASA-Konzept (Context-Aware Service Access) erlaubt es, diese Dienste in Abhängigkeit von der Situation des Nutzers in andere Anwendungen zu integrieren und dort zu nutzen. Diese Integration kann dabei auch durch den Nutzer selbst erfolgen und ermöglicht es so, für jede Situation eines Nutzers ein auf ihn und seine Nutzergruppe abgestimmtes Dienstensemble zu formen, welches ihm in verschiedenen Anwendungen zur Verfügung steht. Neben dem Konzept wird in diesem Beitrag auch auf die bereits durchgeführten und die laufenden CASA-Umsetzungen eingegangen.

## 1 Einleitung

Ein Hochschule verfügt über eine Vielzahl verschiedener Dienste und Anwendungen für ihre Nutzer. Im Bereich der Lehre sind in erster Linie Lernmanagementsysteme (LMS), wie Moodle<sup>1</sup> oder Stud.IP<sup>2</sup> zu nennen, die sowohl von Studenten als auch von Dozenten für die Veranstaltungsverwaltung genutzt werden. Bei Abläufen, wie der Prüfungsanmeldung oder der Abrechnung von Semesterbeiträgen kommen die Studierenden auch mit Anwendungen aus dem Bereich der Hochschulverwaltung, wie HISinOne<sup>3</sup> in Kontakt. Daneben existieren noch diverse anderen Anwendungen, die sowohl zentrale Aufgaben für den Lehrbetrieb erfüllen, z.B. Bibliothekssysteme, als auch speziell von einzelnen Arbeitsgruppen genutzte Dienste wie Terminalserver. Teilweise sind diese Systeme essentiell für den Lehrbetrieb, aber es gibt auch Systeme, die sich eher als Vereinfachung für den Uni-

---

<sup>1</sup><http://www.moodle.de>

<sup>2</sup><http://www.studip.de>

<sup>3</sup><http://www.his.de/abt1/ab01>

versitätsalltag nutzen lassen. Eine auf die jeweiligen Nutzer zugeschnittene Verknüpfung dieser Systeme steht im Fokus der Forschung im Bereich Pervasive University [TL09].

Die effiziente Nutzung dieser Systeme stellt eine Herausforderung dar, die sich auf die Frage zurückführen lässt, wie ein Nutzer in verschiedenen Situationen die relevanten Systeme schnell finden und unkompliziert verwenden kann. Dabei stellt der Aspekt der Situation den Schlüssel zum Problem dar. Wenn bekannt ist, in welcher Situation sich ein Nutzer befindet, ist es möglich zu bestimmen, welche Dienste hier relevant sind. Diese Kontextinformationen lassen sich zusammenfassen als: "Jede Information, die genutzt werden kann um die Situation einer Entität zu charakterisieren. Eine Entität ist dabei eine Person, ein Ort oder Objekt, das als relevant für die Interaktion zwischen einem Nutzer und einer Applikation betrachtet werden kann, einschließlich dem Nutzer und der Applikation selbst." [ADB<sup>+</sup>99]. Die Erfassung dieser Kontextinformationen lässt sich gerade in universitären Umgebungen relativ einfach realisieren [GSB<sup>+</sup>04, BD04]. Neben den klaren Hierarchien und Aufgabenbereichen, in denen sich Studenten und Dozenten befinden, existieren auch durch die LMS bereits Datenquellen, in denen Situationen wie Veranstaltungen beschrieben sind. So ist für einen Nutzer, der im LMS als Student für eine Veranstaltung angemeldet ist, die Prüfungsanmeldung für diese Veranstaltung ein relevanter Dienst. Die Verknüpfung derartiger Dienste und die Integration des einen in den anderen stellt einen der Mehrwerte dar, die durch das CASA Konzept (Context-Aware Service Access) erreicht werden können.

Im den folgenden Abschnitten wird zunächst das CASA Konzept genauer beschrieben. Anschließend wird darauf eingegangen, wie sich die Integration von verschiedenen Diensten in unterschiedliche Anwendungen in Abhängigkeit von der Situation des Nutzers mit Hilfe von CASA realisieren lässt. Der folgende Abschnitt befasst sich mit dem Aspekt der Nutzereinbindung in diese Prozesse, und wie sich durch diese Öffnung sowohl die Qualität, als auch die Quantität der Verknüpfungen erhöhen lassen. Abschließend wird die Evaluation vorgestellt, die aktuell an der Universität Rostock durchgeführt wird und eine Zusammenfassung des gesamten Beitrags geben.

## **2 CASA Konzept**

Das Ziel des CASA Konzeptes ist ein System, mit dem sich unterschiedliche Dienste anhand von Kontextinformationen in verschiedene Anwendungen integrieren lassen. Dabei liegt dem CASA Konzept die These zugrunde, dass sowohl der Nutzer selbst als auch die von ihm verwendeten Anwendungen in der Lage sind die Situation zu beschreiben und zu bestimmen, welche Dienste in dieser Situation relevant sind. Als Dienst wird dabei eine "wiederverwendbare und gekapselte Softwarekomponente mit klar definierter, plattformunabhängiger und lokal sowie über das Netzwerk nutzbarer Schnittstelle" verstanden [Mel07]. Da sich die Kontextinformationen aus mehreren unterschiedlichen Quellen ergeben können war es notwendig ein Konzept zu entwickeln, das in der Lage ist Informationen aus verschiedenen Kontextquellen zu sammeln und zu verarbeiten. Diese Quellen ließen sich im Vorfeld nicht abschließend bestimmen und sind des Weiteren einer ständigen Entwicklung unterworfen, was auch kontinuierliche Anpassungen im CASA System erforderlich

macht. Daher wurde das System so modular aufgebaut, dass ein Kernsystem existiert, in dem die Verarbeitung von Daten in einem vordefinierten Format erfolgt, das mit beliebigen Regeln und Kontextquellen kombiniert werden kann. Dabei sind sowohl die Schnittstellen, als auch die Regeln durch den Nutzer selbst erstellbar und erweiterbar. Diese können dezentral entwickelt werden und auch zwischen den Nutzern ausgetauscht werden um eine schnelle Verbreitung einerseits und andererseits einen hohen Grad an Spezialisierung zu erreichen.

So lassen sich Systeme erstellen, die durch die Verwendung von vielen verschiedenen Kontextinformationen feingranulare Situationen erfassen können und präzise darauf abgestimmte Dienstensembles einbinden können. Im Gegensatz zu einer statischen Kopplung von Diensten, bei der die Dienstensembles zwingend auf die Anwesenheit aller Komponenten angewiesen wäre, erlaubt diese Methode der dynamischen Kopplung auch mit einer Untermenge aller möglichen Kontextquellen zu arbeiten. Als Nachteil des System ist der erhöhte Testaufwand zu werten, der sich durch die Dynamik ergibt, sowie die potentiellen Missbrauchsmöglichkeiten durch die Nutzer, die böswillig das System manipulieren könnten. Als Gegenmaßnahme bieten sich hier Methoden an, mit denen Nutzer fehlerhafte Situationen oder Dienstensembles melden können. Des Weiteren haben sich unterschiedliche Nutzerrollen, Versionierung und Zugriffskontrollen als Sicherungsmechanismen bereits in der Wikipedia bewährt und wären auch hier denkbar.

## **2.1 CASA Knoten**

Das Herz der Architektur stellt der CASA-Knoten dar. Er dient sowohl zur Kontextvereinheitlichung als auch als Verarbeitungseinheit. Ein Knoten besteht dabei aus drei Ebenen, die in Abbildung 1 dargestellt sind. Auf der untersten Ebene stehen die Importer. Dies sind kleine Module, die zum Import von Kontextinformationen dienen. Sie sind in sich geschlossen und speziell für eine Kontextquelle, wie zum Beispiel ein LMS oder ein Zugangssystem, entwickelt. Werden sie aktiviert sammeln sie Informationen aus der Kontextquelle und übersetzen diese Informationen in Fakten, die als normalisierter Kontext in das wissensbasierte System weitergeleitet werden. In dieser mittleren Ebene wird mit Hilfe von Regeln aus den Fakten eine Wissensbasis geformt. Neben diesen Regeln zur Transformation von Kontext existieren auch Regeln für die Dienstausswahl, die auf die Wissensbasis angewendet werden können, um zu bestimmen, welche Dienste aktuell relevant sind. In der obersten Ebene stehen die Aggregationsdienste, die die Schnittstelle zu den Applikationen darstellen. Hier wird die Antwort des wissensbasierten Systems für die Applikation aufbereitet und einem CASA-Plugin in der Applikation übergeben, das für die Darstellung und Einbindung verantwortlich ist.

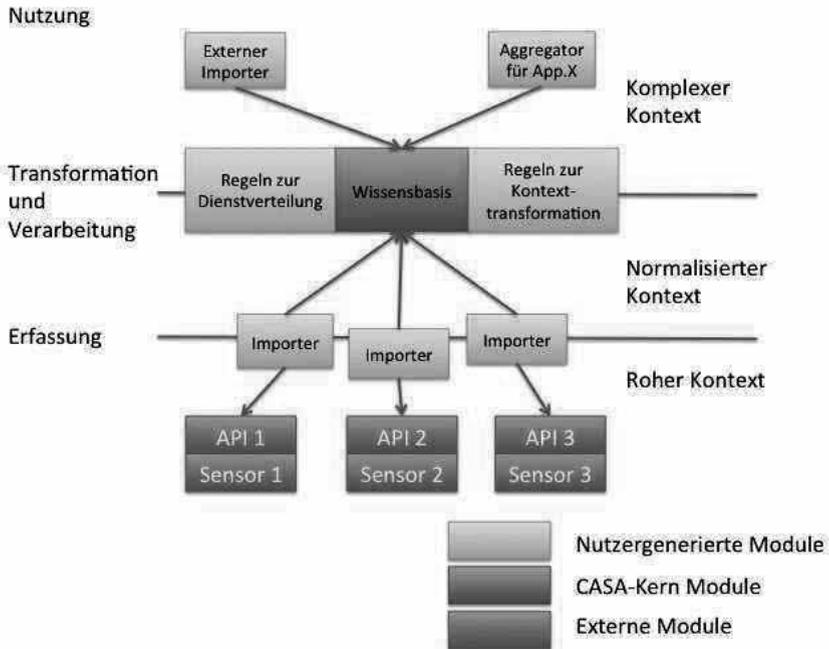


Abbildung 1: Module eines CASA-Knotens

## 2.2 CASA Organisation

Die Kommunikation findet zwischen drei Parteien statt, der Applikation mit dem CASA-Plugin, dem CASA-Knoten und den Diensteanbietern. Es erfolgt keine Kommunikation zwischen der Applikation und den eingebundenen Diensten selbst. Damit folgt die Organisation im CASA Konzept der Idee einer Service-orientierten Architektur mit dem Knoten als Broker[Me107]. Wie in Abbildung 2 dargestellt, stellt eine Applikation, vertreten durch ihr CASA-Plugin eine Anfrage an den Knoten nach aktuell relevanten Diensten. Dabei werden der Anfrage Informationen über die aktuelle Situation der Applikation, wie zum Beispiel der angemeldete Nutzer oder der aktuelle Prozessschritt, beigelegt. Der Aggregatordienst, der im Knoten die Anfrage empfängt wertet sie aus und wendet die entsprechenden Dienstauswahlregeln auf die Wissensbasis an. Die Antwort des Aggregatordienstes enthält Informationen über die relevanten Dienste und wie sie eingebunden werden können. Auf der Seite der Applikation wendet das Plugin diese Informationen an und stellt die Dienste dar.

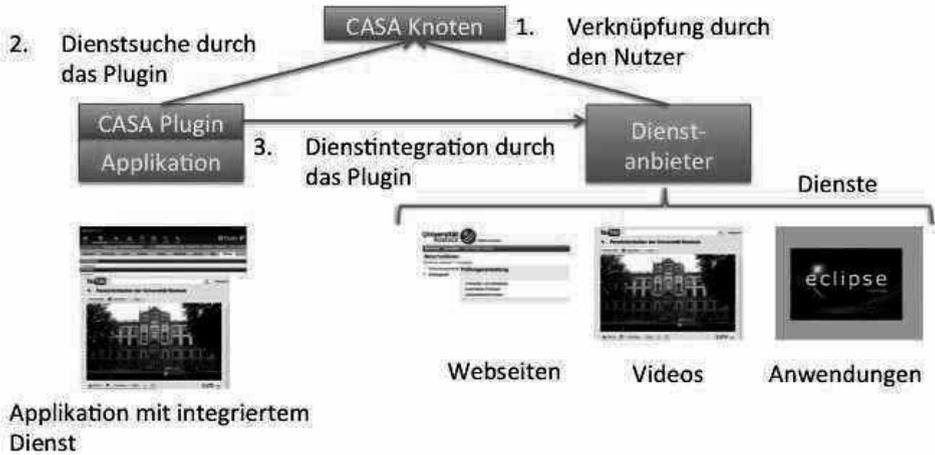


Abbildung 2: Ablauf der Integration eines Dienstes

### 2.3 Kontextuelle Integration von Diensten

Die Nutzung verschiedener CASA-Knoten für eine Dienstausswahl kann diese verfeinern. So kann, wie in Abbildung 3 dargestellt, ein CASA-Knoten mit anderen Knoten verbunden werden. Dabei ist die Dienstausswahl abhängig von den genutzten Kontextquellen. Eine Integration mit Kontextinformationen ohne Bezug zur aktuellen Situation des Nutzers führt zu einer nutzerunspezifischen Dienstausswahl. Erfolgt die Integration mit Hilfe von gruppenbasierten Kontextquellen, wie zum Beispiel Gruppen-Kalendern, ist die Auswahl bereits gruppenspezifisch. Die von dem öffentlich genutzten Knoten bereits normalisierten Kontextinformationen können hier durch den Knoten angefragt werden, der durch die Gruppe genutzt wird. Die Informationen aus dem Gruppen-Knoten stehen dem öffentlichen Knoten jedoch nicht zur Verfügung. Analog verhält es sich zwischen einem privat genutzten und einem durch eine Gruppe genutzten Knoten. So kann gewährleistet werden, dass Informationen aus dem privaten oder gruppenorientierten Bereich diese nicht verlassen. Neben den Kontextinformationen können so auch Dienstvorschläge aus verschiedenen Knoten aggregiert und gesammelt für einen Nutzer angeboten werden.

### 2.4 Nutzergetriebene Integration von Diensten

Während die kontextbasierte Integration mit einzelnen Knoten realisierbar ist, ist es auch möglich mehrere Knoten miteinander zu verknüpfen. Wie bereits in Abbildung 1 dargestellt, ist es möglich, dass der Knoten vom Nutzer selbst durch eigene Regeln, Dienste und Kontextquellen erweitert wird. Dabei wird die Dienstausswahl an die Bedürfnisse

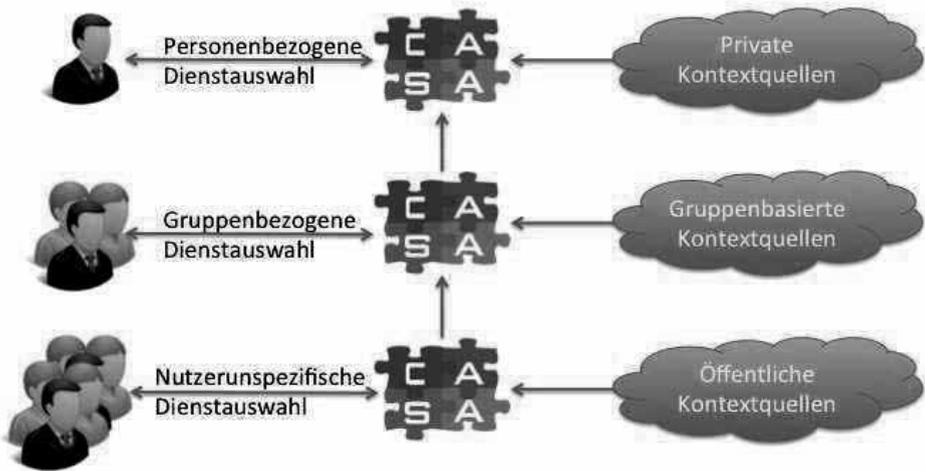


Abbildung 3: Kaskadierung mehrerer Knoten zur Verfeinerung der Dienstauswahl

des Nutzers, beziehungsweise seiner Gruppe angepasst. Dies wird ermöglicht, indem das Kontextmodell des CASA-Knotens eine grundlegende Ordnung definiert, die durch den Nutzer erweitert werden kann. So wurde für einen CASA-Knoten, der für die Einbindung von Diensten in ein LMS konfiguriert ist, das Modell um Konzepte wie *Veranstaltung* und *Raum* erweitert. Diese Konzepte bilden die Grundlage für die Regeln, die ein Nutzer für diesen Knoten formulieren kann. Da in anderen Umgebungen ein anderes Verständnis für das Konzept *Veranstaltung* existieren kann, ist es notwendig die Konzepte zwischen zwei CASA-Knoten anzupassen. Dies erfordert es, dass auch die Importer durch den Nutzer erstellt und erweitert werden können. So kann bei Bedarf das Konzept *Veranstaltung*, das eigentlich eine Reihe von Terminen umfasst mit dem Konzept einer Veranstaltung aus einem Kalender verknüpft werden, in dem jeder Eintrag für genau eine Veranstaltung steht.

### 3 Implementierung des CASA Systems

Für die Evaluation der kontextbasierten Dienstintegration wurde das CASA-Konzept prototypisch umgesetzt. Da das Lernmanagementsystem Stud.IP bereits viele Informationen erfasst, und es eine Standardanwendung für Studenten sowie Dozenten darstellt, wurde es sowohl als Anwendung als auch als eine der Kontextquellen gewählt. Stud.IP selbst ist ein Open Source System, das an über 50 Hochschulen mit über 450.000 registrierten Nutzern zum Einsatz kommt. Als Dienstanbieter wurde eine an der Universität Rostock vorhandene intelligente Umgebung gewählt. In diesem Spezialseminarraum ist es möglich über eine graphische Benutzerschnittstelle Beamer, Sonnenblenden, Leinwände und ande-

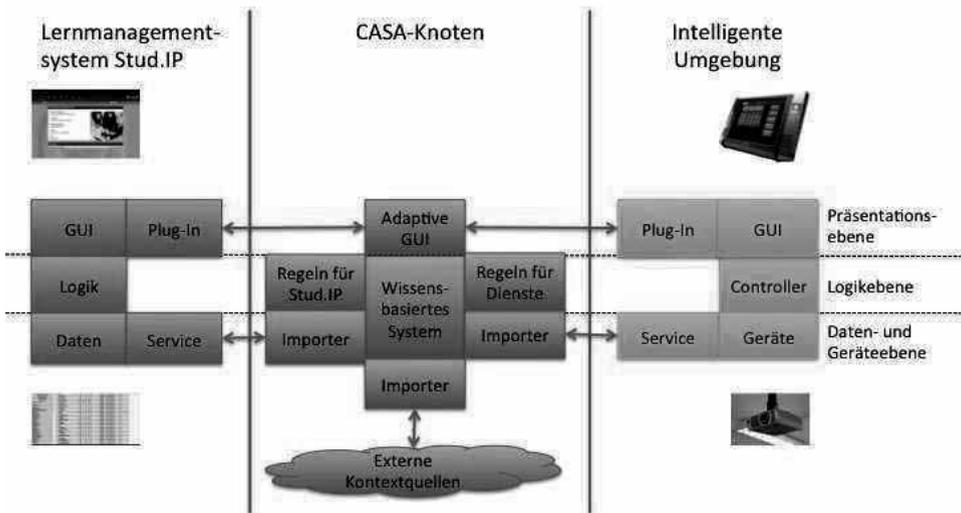


Abbildung 4: Implementierung für das Lernmanagementsystem Stud.IP

re installierte Geräte zu steuern. Da diese Steuerung sehr komplex sein kann, wurde die Idee entwickelt CASA dafür einzusetzen, dass sowohl Studenten, als auch Dozenten ohne Erfahrung mit den Diensten des Raumes diese nutzen können. Dazu wurden einerseits Regeln entwickelt, die Dienst-Konfigurationen für die Lehrszenarien in dem Raum enthalten, und andererseits Regeln, die ermitteln, wann so eine Situation vorliegt. Die Fakten werden über Importer aus dem Stud.IP gesammelt und das Plugin übermittelt in dem Moment, wo eine Veranstaltungsseite für diesen Raum aufgerufen wird, die Nutzerrolle an das CASA System. Anschließend wird evaluiert, welche Dienste durch diese Nutzerrolle genutzt werden dürfen. Diese Dienste werden dann über eine adaptive GUI in das Stud.IP eingebunden und können so direkt von dort aus genutzt werden.

Neben der Einbindung weiterer statischer Webseiten, wie der Darstellung eines Buchkapitels aus einem für die Vorlesung relevanten Buches sowie einer Suchanfrage im Bibliothekssystem nach vorlesungsspezifischen Stichwörtern, wurde auch eine dynamische Haltestellensuche implementiert. Dabei wird die Gebäudebezeichnung der aktuellen Veranstaltung als Parameter übergeben und über Daten aus dem Rostocker OpenData-Projekt<sup>4</sup> die geographische Position ermittelt. Verbunden mit den Haltestelleninformationen aus der gleichen Quelle und einem Verzeichnis der Internationalen Bahnhofnummern lassen sich direkte Links zu den jeweils zum Veranstaltungsort nächstgelegenen Haltestellen generieren.

<sup>4</sup><http://www.opendata-hro.de>

Für das wissensbasierte System wurde Drools<sup>5</sup> verwendet. Es ist ebenfalls Open Source und in Java implementiert. Des Weiteren verfügt es über eine aktive Gemeinschaft und wird auch weiterentwickelt. Die Stud.IP-Konzepte, wie Lehrveranstaltungen und Nutzerrollen, wurden in eine generische objektorientierte Ontologie übersetzt und stellen damit die Erweiterung der bereits erwähnten Grundmodells dar. Die Importer, sowie die Web Services sind ebenfalls in Java umgesetzt und erlauben so eine einfache Wiederverwendung. Als Problem ist festzuhalten, dass Dienste wie das Buchkapitel des Springer-Verlags nur aus dem Netz der Universität frei zugreifbar sind und Nutzer, die das System über UMTS in einer Vorlesung nutzen hier nur eingeschränkten Zugriff haben. Des Weiteren mangelt es dem System aktuell noch an einer Umsetzung für Single Sign-on, was dazu führt, dass ein Nutzer sich auf der Prüfungsseite, die in eine Veranstaltungsseite eingebunden wurde, erneut anmelden muss. Ein Ansatz hierzu wurde zwar in [LLT11] konzipiert und prototypisch für ein vergleichbares Szenario umgesetzt, jedoch erfordert die hier dargestellte Lösung sehr tiefgreifende Zugriffe auf die Datenbanken des Studip.

## 4 Aktuelle Evaluationen an der Universität Rostock

Um auch die Aspekte der nutzergetriebenen Dienstintegration evaluieren zu können wird aktuell eine noch breitere Integration in das Stud.IP der Universität Rostock vorbereitet. Dazu wurde das Plugin im Stud.IP so erweitert, dass es den Nutzern erlaubt selbstständig Dienste in das System einzutragen und mit Veranstaltungen zu verknüpfen. So soll überprüft werden, ob sich die Nutzer durch diese Möglichkeiten mehr mit den Veranstaltungen befassen und ob sich durch die integrierten Dienste die wahrgenommene Qualität der Veranstaltungsseiten erhöht. Des Weiteren wird in einer laufenden studentischen Arbeit überprüft, ob sich auch durch den Einsatz von Gamification, also der Verwendung von spieletypischen Mechanismen in spielfremden Kontexten, die Nutzerbeteiligung erhöhen und die Qualität verbessern lässt. Nebenbei stellt diese Evaluation auch ein Performance-Test dar, da hier eine deutlich höhere Zahl an Nutzerinteraktionen als bisher auftreten wird. Somit wird sich mit dieser Evaluation auch zeigen, ob die basierend auf Überlegungen gewählten Software-Produkte in der Praxis geeignet sind. Ergebnisse hierzu sind jedoch erst im Sommersemester 2013 zu erwarten.

## 5 Zusammenfassung

An Hochschulen sind heute bereits viele Prozesse über Webschnittstellen verfügbar. Diese reichen von der Prüfungsanmeldung, über die Bereitstellung von Lehrmaterial bis hin zur Fernleihe. Um diese unterschiedlichen Inhalte für den Nutzer leicht und situationsbezogen erreichbar zu machen wurde in diesem Beitrag das CASA-Konzept zur nutzergetriebenen, kontextbewussten Integration von Diensten in Anwendung vorgestellt. Dazu wurde neben der Struktur des CASA-Knoten auch die Verknüpfung mehrerer Knoten erläutert.

---

<sup>5</sup><http://www.jboss.org/drools>

Anschließend wurden die Anwendungsfälle der Integration von Diensten auf Basis von Situationsinformationen, sowie die Integration von Diensten durch den Nutzer selbst vorgestellt. Abschließend wurde ein Einblick in die bereits umgesetzte Evaluation des ersten Anwendungsfalls und ein Ausblick auf weitere Evaluationen gegeben.

## Danksagung

Die Arbeit von Philipp Lehsten und Djamshid Tavangarian wird durch die Deutsche Forschungsgemeinschaft (DFG), im speziellen durch das Graduiertenkolleg 1424 MuSAMA, gefördert. Die Autoren bedanken sich ebenfalls bei den Studenten Martin Hölzel und Stefan Wendt für ihre Unterstützung bei der Umsetzung des Systems.

## Literatur

- [ADB<sup>+</sup>99] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith und P. Steggles. Towards a Better Understanding of Context and Context-Awareness. In Hans-W. Gellersen, Hrsg., *Handheld and Ubiquitous Computing*, Jgg. 1707 of *Lecture Notes in Computer Science*, Seiten 304–307. Springer Berlin / Heidelberg, 1999.
- [BD04] L. Barkhuus und P. Dourish. Everyday Encounters with Context-Aware Computing in a Campus Environment. In *UbiComp 2004: Ubiquitous Computing*, Jgg. 3205 of *Lecture Notes in Computer Science*, Seiten 232–249. Springer Berlin / Heidelberg, 2004.
- [GSB<sup>+</sup>04] W.G. Griswold, P. Shanahan, S.W. Brown, R. Boyer, M. Ratto, R.B. Shapiro und T.M. Truong. ActiveCampus: experiments in community-oriented ubiquitous computing. *Computer*, 37(10):73 – 81, oct. 2004.
- [LLT11] U. Lucke, P. Lehsten und D. Tavangarian. Conceptual Integration of Security Management in Designing Context-aware Environments. In *Second Workshop on Context-Systems Design, Evaluation and Optimisation (CoSDEO 2011) at 24th Int. Conference on Architecture of Computing Systems (ARCS)*, Como, 2011.
- [Mel07] I. Melzer. *Service-orientierte Architekturen mit Web Services - Konzepte, Standards, Praxis (2. Aufl.)*. Spektrum Akademischer Verlag, 2007.
- [TL09] D. Tavangarian und U. Lucke. Pervasive University - A Technical Perspective (Die Pervasive University aus technischer Perspektive). *it - Information Technology*, 51(1):6–13, 2009.

