Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Modellierung des Accountmanagement-Systems der Fakultät für Informatik

Studienarbeit am Institut für Telematik in Kooperation
mit der Abteilung technische Infrastruktur (ATIS)
Forschungsbereich Dezentrale Systeme und Netzdienste
Prof. Dr. Hannes Hartenstein
Fakultät für Informatik
Universität Karlsruhe (TH)

von

Christoph Zipperle

Betreuer:
Dipl.-Inf. Thorsten Höllrigl
Dipl.–Math. Klaus Scheibenberger

Tag der Anmeldung: 03.12.07
Tag der Abgabe: 29.02.08

Ehrenwörtliche Erklärung

Ich erkläre hiermit, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

Karlsruhe, den 13.02.2008

Christoph Zipperle

Inhaltsverzeichnis

1 Einleitung

Der Leser wird zunächst in das Themengebiet „Identity Management“ eingeführt. Nach der Erläuterung der organisatorischen Gliederung an der Universität Karlsruhe sowie eines Beispielszenarios folgt eine kurze Übersicht der Arbeit, welche die Aufgabenstellung beinhaltet.

1.1 Einführung in das Themengebiet

Jede Person besitzt eine Identität. Diese Identität zeichnet sich durch eine Menge von Attributen aus, welche in ihrer Summe die Person eindeutig identifizierbar machen. In der Regel können mittels der Attribute Vorname, Nachname, Geburtsdatum und Geburtsort bereits Personen innerhalb einer Gruppe eindeutig bestimmt werden. Dennoch umfasst eine Identität weitaus mehr Attribute, als die eben genannten.
Identitäten können mit Hilfe von Rollen konzeptionell zusammengefasst werden. So erhält eine Person z.B. durch das Unterzeichnen eines Arbeitsvertrags die Rolle eines Arbeitnehmers, durch das Immatrikulieren an einer Universität die Rolle eines Studenten. Durch diese Rolle wird die Identität um spezifische Attribute (z.B. Telefonnummer im Büro, Matrikelnummer) und Rechte (z.B. Zugang zu gewissen Akten oder Räumlichkeiten) erweitert. Rollen werden also Identitäten zugewiesen, die dann die rollenspezifischen Merkmale für die Dauer der Zuweisung übernehmen.

Arbeitsabläufe mit denen Universitäten (oder Organisationen) ihre Geschäftsziele gegenüber Kunden verfolgen, nennt man Geschäftsprozesse. Diese beschreiben verwaltende sowie organisatorische Abläufe. Geschäftsprozesse sind zunehmend IT-basiert, d.h. viele dieser (Teil-)Prozesse laufen auf Computersystemen ab oder werden durch sie bei der Ausführung unterstützt [HS+07]. Die IT-basierten (Teil-)Abläufe der Geschäftsprozesse greifen für ihre Durchführung auf digitale Repräsentationen von Identitäten, den digitalen Identitäten, zu. Eine digitale Identität wird durch (digital gespeicherte) Attribute gebildet, die eine Person innerhalb einer Organisation eindeutig identifizieren. Mittels des Konzeptes von (digitalen) Rollen lassen sich digitale Identitäten anforderungsabhängig strukturieren.

Um IT-Dienste (z.B. E-Mail, Dateiserver) und Arbeitsplätze tatsächlich nutzen zu können, müssen den Mitarbeitern bzw. deren digitalen Identitäten technische Attribute, die von den dienstrelevanten Systemen ausgewertet und verarbeitet werden können, zugeordnet werden. Ein einfaches Beispiel ist das Attribut „Passwort“, das zur Authentifizierung benötigt wird. Die Menge der für die Nutzung der Dienste notwendigen Attribute werden in Nutzerkonten (Accounts) zusammengefasst, die dann, beispielsweise über einen Verzeichnisdienst (s. Kapitel 2.2) den dienstrelevanten Systemen bereitgestellt werden.

Das Verwalten der digitalen Identitäten sowie das Zuweisen von Rollen und Accounts zu diesen Identitäten ist Aufgabe des Identity Managements (IDM) [KR+03]. Ein Teilgebiet des IDM, dessen Zielsetzung das Verwalten der Accounts ist, wird Account Management (AM) genannt. Hierbei steht der Account als zu verwaltendes, systemnahes Objekt im Vordergrund, wohingegen beim IDM die von Systemen und

Accounts entkoppelte Identität im Fokus steht. Es lässt sich jedoch nur schwer eine strikte Grenze zwischen IDM und AM ziehen. Die zentralen Themen des AM (und damit auch Themen des IDM) sind Zugriffskontrolle mittels Authentifizierung und Autorisation [HS+07]. Die Einhaltung der gesetzlich vorgegebenen Datenschutzrichtlinien ist sowohl beim IDM als auch beim AM obligatorisch [KR+03]. IDM-Systeme bezeichnen Softwaresysteme mit zugehöriger Infrastruktur zur Bewältigung der Aufgaben des IDM. Analog dazu verhält es sich mit AM-Systemen.

1.2 Beispielszenario

1.2.1 Organisationsstruktur des IDM der Universität Karlsruhe

Das IDM-System der Universität Karlsruhe wird von der Forschungsgruppe „Karlsruher Integriertes InformationsManagement“ (KIM) im Rechenzentrum (RZ) betrieben. Betrachtet man die Universität als eine Stern-Topologie mit dem KIM-IDM in der Mitte, dann sind die äußeren Knoten die organisatorischen Einheiten, die für die in ihrem Zuständigkeitsbereich erforderliche Verwaltung von Nutzerkonten selbst verantwortlich sind. Diese Einheiten bekommen die Identitätsdaten vom KIM-IDM über wohldefinierte Schnittstellen bereitgestellt und werden im Kontext dieser Arbeit als Satelliten bezeichnet (siehe Abbildung 1). [HS+06]
Dazu gehört neben dem Rechenzentrum und weiteren Einheiten auch die Abteilung technische Infrastruktur (ATIS), die zuständig für das IDM bzw. das AM der Fakultät für Informatik ist. Das AM dieser Fakultät ist Thema der vorliegenden Arbeit. Details zur Übergabe der Identitäten vom KIM-IDM an die Satelliten sind in Kapitel 4 erläutert.

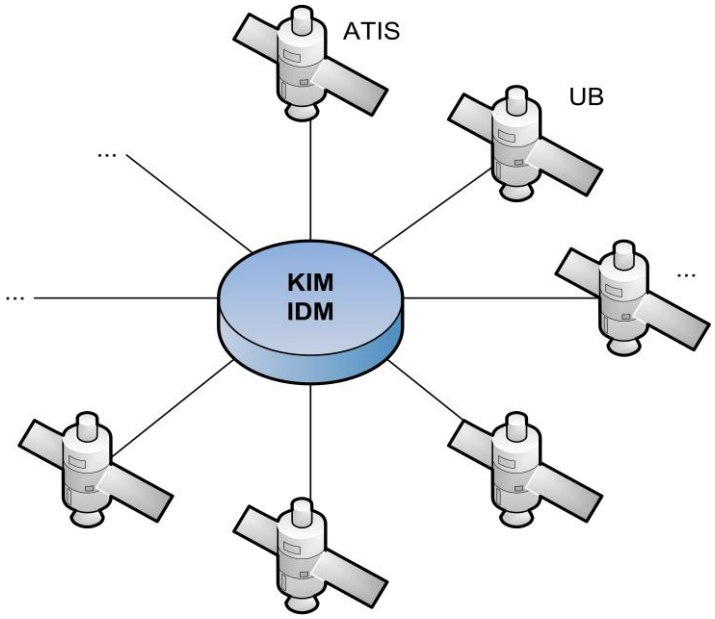

Abbildung 1: KIM-IDM

Die Fakultät für Informatik gliedert sich weiter in Einrichtungen. Diese sind zum Beispiel Institute wie das Institut für Dialog- und Betriebssysteme (IDBS), der

Poolraum für die Studenten oder die Informatik-Bibliothek. Institute wie das IDBS untergliedern sich weiter in Einrichtungen, wie z.B. Forschungsbereiche (siehe Abbildung 2).

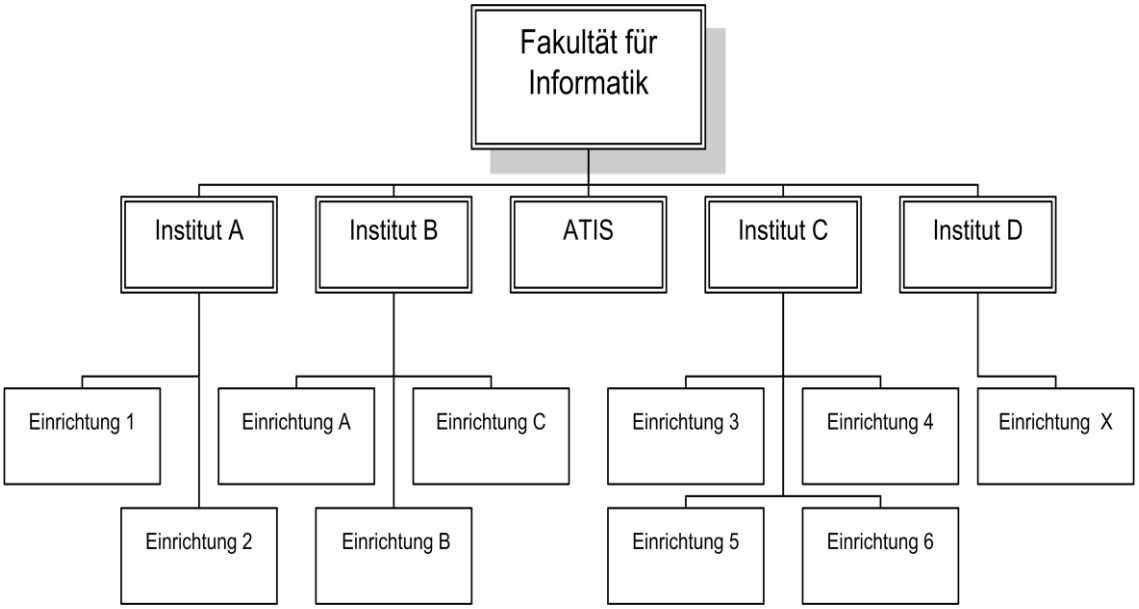

Abbildung 2: Gliederung der Fakultät

Jede dieser Einrichtungen vergibt für die ihr zugehörigen Benutzer (Studenten oder Mitarbeiter) Accounts, damit diese sich an Arbeitsplätzen anmelden und lokale -also von den Einrichtungen angebotene- Informations- und Kommunikations-Dienste (IuK-Dienste) nutzen können. Zentrale Dienste wie E-Mail oder Virtual Private Network (VPN) werden von der ATIS angeboten. Bisher sind die lokalen und zentralen Dienste voneinander unabhängig. Die lokalen Dienste bekommen ihre Attribute durch die lokalen Nutzerkonten bereitgestellt, die Attribute der zentralen Dienste werden durch das bestehende AM-System der ATIS verwaltet und bereitgestellt. Zukünftig soll die Möglichkeit geschaffen werden sowohl die lokalen als auch die zentralen Nutzerkonten über eine von der ATIS zentral betriebene, aber dezentral (von den einzelnen Einrichtungen) verwaltete Struktur bereitzustellen.

Die von einem der Satelliten benötigten Identitätsdaten, beispielsweise Name, Vorname, Studienfach und Immatrikulationsstatus, werden zwischen dem Satelliten und der, für diese Daten, autoritative Quelle vereinbart. Für die beispielhaft genannten Identitätsdaten wäre dies die Universitätsverwaltung. Aus datenschutzrechtlichen Gründen spielt für die Vereinbarung u.a. der Zweck, d.h. warum die Daten benötigt werden, eine entscheidende Rolle. Das KIM-IDM erhält diese Daten dann von der Verwaltung der Universität, um diese über eine wohldefinierte Schnittstelle an den Satelliten zu provisionieren. In der Folge werden auch Änderungen der Daten zeitnah an den Satelliten übermittelt und man erreicht somit eine hohe Aktualität der Daten, z.B. hinsichtlich des aktuellen Immatrikulationsstatus eines Studierenden, bei dem entsprechenden Satelliten.

Das AM der Fakultät für Informatik ist Thema der vorliegenden Arbeit. Details zur Übergabe der Identitätsdaten vom KIM-IDM an die Satelliten sind in Kapitel 4 erläutert.

1.2.2 Identitäten - Accounts

Im Folgenden wird ein Student der Fakultät betrachtet:
Dieser wird durch seinen vollen Namen, Geburtsort und Geburtstag in der Regel eindeutig identifiziert. Diese Informationen wurden bei der Immatrikulation von der Verwaltung digital erfasst und bilden mit den restlichen, digital erfassten Eigenschaften seine digitale Identität. Innerhalb der Universität hat er die Rolle Student. Damit sämtliche verwaltungstechnischen sowie studentischen Geschäftsprozesse der Universität reibungslos mit dieser Identität umgehen können, muss diese um das rollenspezifische (studentische) Attribut Matrikelnummer erweitert werden. Erst durch die Matrikelnummer, die er bei der Immatrikulation erhält, kann er innerhalb der Universität Karlsruhe eindeutig identifiziert werden. Ein weiteres studentisches Attribut ist der Immatrikulationsstatus. Dieses Attribut ist für die Gültigkeit der studentischen Accounts von Bedeutung. Mitarbeiter der Universität besitzen diese Attribute nicht, da sie speziell für die Geschäftsprozesse, welche die Studenten betreffen, von Bedeutung sind. Die Mitarbeiter haben andere, mitarbeiterspezifische Attribute wie z.B. Kontodaten, die für Geschäftsprozesse der Mitarbeiter (z.B. Gehaltsabrechnung) von Bedeutung sind.
Aus Sicht der ATIS ist die hier betrachtete Person sowohl Student als auch Mitarbeiter – er ist als wissenschaftliche Hilfskraft angestellt. Aus diesem Grund erhält sie einen Studenten-Account und einen Mitarbeiter-Account mit jeweils zugeschnittenen Eigenschaften. Jeder dieser Accounts erweitert die Identität der Person um bereichsspezifische Attribute. Der studentische Account innerhalb der Fakultät für Informatik erweitert die digitale Identität beispielsweise um das Attribut Druckguthaben, durch das der Student berechtigt wird, die Drucker im ATIS-Poolraum zu nutzen. Durch den Mitarbeiter-Account erhält die Person im Zuständigkeitsbereich der ATIS die Berechtigungen und Attribute eines Mitarbeiters, und seine digitale Identität wird um Mitarbeiter-relevante Attribute erweitert.
Ähnlich verhält es sich in den Bereichen anderer Satelliten, z.B. im Kontext der Universitätsbibliothek (UB). Auch hier besitzt jeder Student einen Account, um die Dienste der UB nutzen zu können. Dem Account werden alle Attribute, die für die Geschäftsprozesse der UB benötigt werden (z.B. „Liste der ausgeliehenen Bücher“), hinzugefügt. Accounts fassen also im Bereich eines Satelliten, auf einer bestimmten Ebene, systemtechnisch erforderliche Attribute zusammen und sind dabei mit einer Identität verknüpft.

1.3 Aufgaben und Struktur

Aufgabe dieser Studienarbeit ist der Entwurf eines neuen Systems für das Account-Management an der Fakultät für Informatik. Dieses soll das bestehende Account-Management-System (im Folgenden als AM-System bezeichnet) der ATIS, inklusive der derzeit eingesetzten Verwaltungs-Software UserADM ablösen und darüber hinaus zusätzliche Funktionalität (z.B. Selbstbedienungsfunktionalität) anbieten.

Um ein größtmögliches Verständnis des Lesers zu erzielen, wird dieser in Kapitel 1 mit dem Thema Identity Management und den organisatorischen Gegebenheiten an der Universität Karlsruhe vertraut gemacht, bevor in Kapitel 2 (Technische Grundlagen) die eingesetzte Modellierungssprache und weitere, für diese Studienarbeit relevante, Technologien vorgestellt werden. Der Entwurf komplexer Systeme verlangt eine Anforderungsanalyse (p.98f)[Ba00], dazu wird zunächst das bestehende AM-System inklusive des UserADM erfasst und analysiert (Kapitel 3 – Analyse des bestehenden Systems). Die so gewonnenen Anforderungen und Erfahrungen werden zusammen mit neuen Ideen und Konzepten in Kapitel 4 (Anforderungsanalyse des neuen Systems) ausgearbeitet, strukturiert und übersichtlich dargestellt. Die aus Kapitel 4 gewonnenen Kenntnisse werden bei der Modellierung der neuen Verwaltungssoftware (Kapitel 5 - Modellierung der Anwendungsfälle) als Grundlage genutzt. Zur Modellierung gehören:

- Das Erfassen aller Anwendungsfälle
- Das Erfassen aller Akteure
- Die Gliederung aller Anwendungsfälle mittels Anwendungsfall-diagrammen

Beispielhaft wird der komplexe Anwendungsfall „Account anlegen“ detailliert untersucht und modelliert. Dies beinhaltet:

- Die textuelle Beschreibung des Anwendungsfalls „Account anlegen“
- Konkrete Modellierung zum Anwendungsfall „Account anlegen“

Dabei sind die vorgegebene Zielarchitektur, sowie die vom KIM-IDM-System gegebenen Schnittstellen zu beachten. Das so entstehende Modell soll als fundierte Basis für die spätere Implementierung der Software dienen. Abschließend wird die Arbeit nochmals zusammengefasst, resümiert und mit einem Ausblick abgeschlossen (Kapitel 6 – Zusammenfassung und Ausblick).

2 Technische Grundlagen

Um dem Leser die zum Verständnis der Studienarbeit nötigen Grundlagen zu schaffen, werden in diesem Kapitel Technologien vorgestellt, die für diese Studienarbeit und das zu modellierende AM-System essentiell sind. Einerseits ist dies die Unified Modeling Language, mit der sowohl die Infrastruktur, als auch die Management-Anwendung des neuen Systems modelliert werden. Zum anderen sind es Web Services, mit denen die Verbindung zwischen dem AM-System und dem KIM-IDM-System realisiert wird, und das Lightweight Directory Access Protocol, das zur Datenhaltung im AM-System eingesetzt wird.

2.1 Unified Modeling Language (UML)

„Die Unified Modeling Language ist eine Sprache und Notation zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme“ [Oe05]. Sie wird eingesetzt, um den Entwurf von (komplexer) Software zu unterstützen. UML entstand aus vielen verschiedenen Entwurfsmethoden, die nach dem Aufkommen objektorientierter Programmiersprachen wie Smalltalk unabhängig voneinander entwickelt wurden. UML schaffte es als erste Modellierungssprache, die wichtigsten dieser Methoden zu vereinen und -sofern das möglich war- allen Anforderungen der Softwareentwickler gerecht zu werden. Version 1.0 wurde im Jahr 1997 veröffentlicht, noch im selben Jahr folgte Version 1.1, welche im November 1997 von der Object Management Group [OMG] als Standard anerkannt wurde. Eine generelle, kurze Einführung in die Konzepte von UML vermittelt [Be03]. Version 2.0, eine umfassende Überarbeitung mit vielen neuen Eigenschaften und gravierenden Änderungen, wurde 2005 von der OMG anerkannt. Eine Zusammenfassung der wichtigsten Veränderungen in Version 2.0 liefert [Se05]. Es folgt eine kurze Übersicht der wichtigsten UML-Konzepte:

UML kennt drei Arten von Bausteinen:

- Dinge: Abstraktionen und damit wichtigste Anteile eines Modells
- Beziehungen: Sie verbinden einzelne Dinge miteinander
- Diagramme: Sammlungen von Dingen und Beziehungen

Dinge sind unterteilt in 4 Gruppen:

- Strukturdinge
- Verhaltensdinge
- Gruppierungsdinge
- Anmerkungsdinge

Auch Beziehungen lassen sich in 4 Gruppen einteilen:

- Abhängigkeiten
- Assoziationen
- Generalisierungen
- Realisierungen

Diagramme erhält man durch das Kombinieren von Dingen und Beziehungen nach den vorgegebenen Regeln der UML. Insgesamt gibt es 13 verschiedene Diagrammtypen:

- Klassendiagramm
- Objektdiagramm
- Komponentendiagramm
- Montagediagramm
- Anwendungsfalldiagramm
- Sequenzdiagramm
- Kommunikationsdiagramm
- Zustandsdiagramm
- Aktivitätsdiagramm
- Verteilungsdiagramm
- Paketdiagramm
- Zeitverlaufsdiagramm
- Interaktionsübersichtsdiagramm

UML 2.0 ist eine sehr komplexe Modellierungssprache, deren komplettes Regelwerk über 1000 gedruckte Seiten füllt. Allerdings reichen 20-30 % der UML-Elemente in 80% aller Fälle aus, um das gewünschte Ergebnis zu erzielen. (p.13)[Oe05]
Für diese Studienarbeit werden wir die im folgenden Abschnitt vorgestellten Diagramme zum Modellieren einsetzen. Als Referenz wurde (p.51f)[BR+06] benutzt:

Klassendiagramm

Bei der Modellierung von objektorientierten Systemen ist das Klassendiagramm der am häufigsten benutzte Diagrammtyp. Die einzelnen Klassen, Interfaces und Kollaborationen werden zusammen mit den untereinander herrschenden Beziehungen dargestellt. Sie bieten also eine statische Sicht auf den Entwurf. Klassen werden dargestellt als Rechtecke und ihre Beziehungen als Linien.

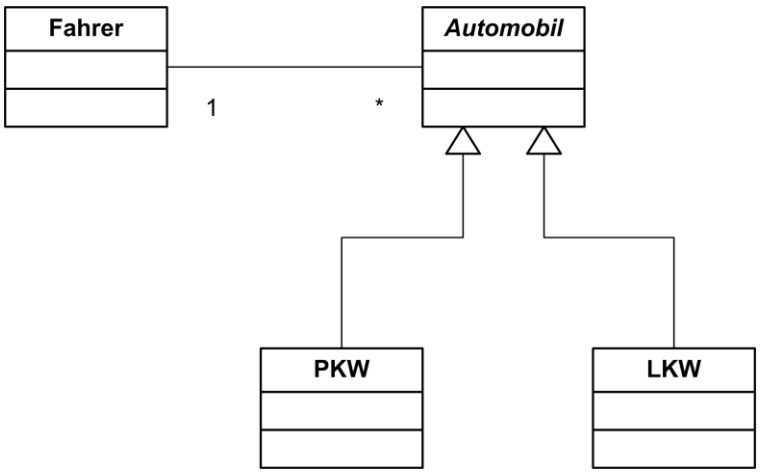

Abbildung 3: Beispiel Klassendiagramm

Komponentendiagramm

Das Komponentendiagramm ist eine Variante des Klassendiagramms, welche auch die statische Sicht der Entwurfsimplementierung eines Systems beschreibt. Allerdings werden die Klassen hier zusammen mit Interfaces und Ports in Komponenten gekapselt, welche wiederum verschachtelt dargestellt werden können. Untereinander kommunizieren Komponenten über bereitgestellte bzw. benötigte Schnittstellen.
Dieser Diagrammtyp eignet sich sehr gut zum Aufbau großer Systeme aus kleinen Bestandteilen bzw. zum Erstellen der Übersicht eines großen Systems.

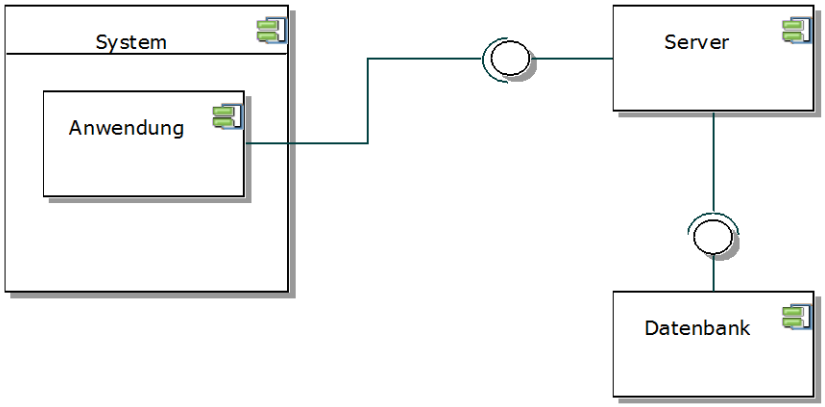

Abbildung 4: Beispiel Komponentendiagramm

Anwendungsfalldiagramm

In Anwendungsfalldiagrammen wird das Verhalten von Systemen durch seine Anwendungsfälle beschrieben. Zusätzlich werden diese Anwendungsfälle mit Akteuren (eine besondere Art der Klasse) in Beziehungen gesetzt. Wie auch die vorigen Diagramme bietet es eine statische Sicht auf das System.

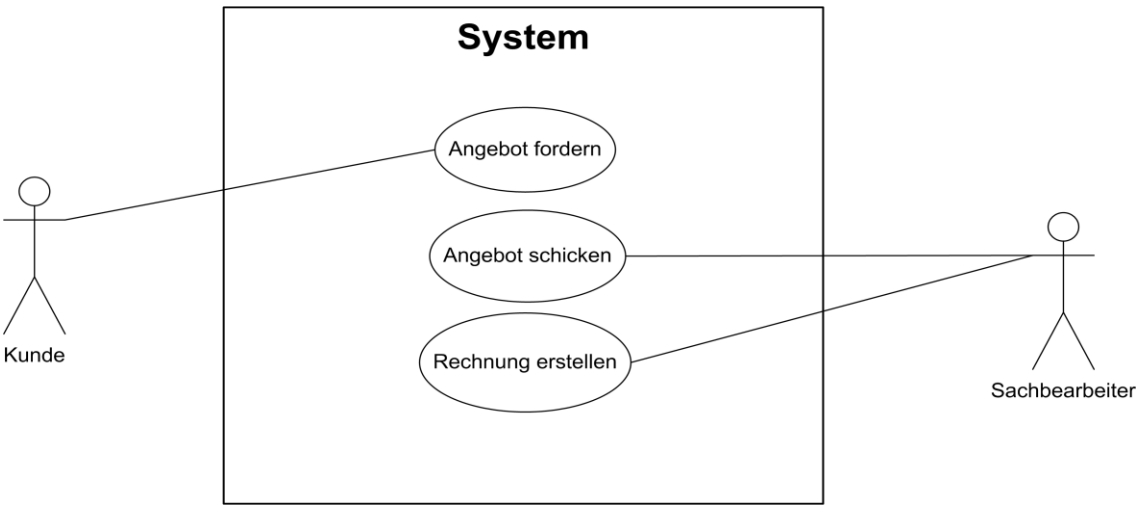

Abbildung 5: Beispiel Anwendungsfalldiagramm

Aktivitätsdiagramm

Aktivitätsdiagramme beschreiben den Kontroll- und Datenfluss innerhalb eines Anwendungsfalls bzw. (Geschäfts-)Prozesses. Es handelt sich dabei um dynamische Modelle. Im Rahmen dieser Studienarbeit werden sie dazu verwendet, einzelne Anwendungsfälle detailliert zu beschreiben.

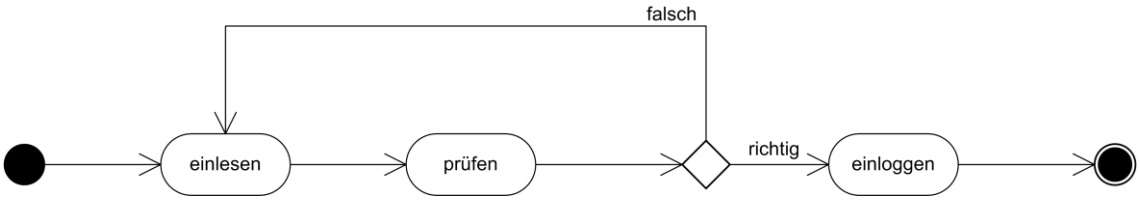

Abbildung 6: Beispiel Aktivitätsdiagramm

Sequenzdiagramm

Sequenzdiagramme gehören zur Klasse der Interaktionsdiagramme. Sie ermöglichen die Darstellung der Interaktion zwischen Objekten und/oder Klassen sowie der ausgetauschten Nachrichten. Es handelt sich also ebenfalls um ein dynamisches Diagramm. Durch die detaillierte Beschreibung der Objekte und der ausgetauschten Nachrichten kann der Programmierer Sequenzdiagramme direkt in Code umsetzen.

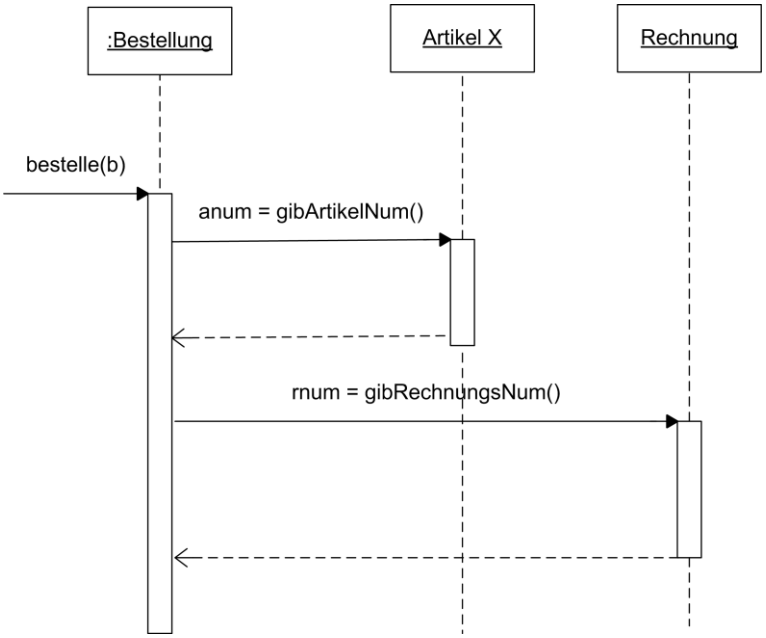

Abbildung 7: Beispiel Sequenzdiagramm

Um ein tieferes Verständnis für UML zu erlangen werden die schon in diesem Abschnitt referenzierten Bücher [BR+06] und [Oe05] empfohlen.

2.2 Lightweight Directory Access Protocol (LDAP)

Das Lightweight Directory Access Protocol ist ein Protokoll zum Zugriff (Abfrage und Modifikation) auf Verzeichnisdienste. Es befindet sich im TCP/IP Protokollstapel auf der Anwendungsebene und ist zuständig für die Steuerung der Kommunikation zwischen dem LDAP-Client und dem Directory Server. Es handelt sich also um eine Kommunikation nach dem Vorbild des Client-Server-Modells. Im Server selbst werden objektspezifische Daten hierarchisch (in einer Baumstruktur) abgelegt. Die Kommunikation erfolgt anhand von Abfragen.
Seinen Ursprung hat LDAP im Directory Access Protocol (DAP), welches als Teil des sehr umfangreichen X.500 Standards entwickelt wurde. Das größte Problem von DAP, neben seiner Komplexität, ist, dass es auf dem kompletten ISO/OSI Protokollstapel aufsetzt, was eine Implementierung sehr aufwendig werden lässt. Mit LDAP verfolgte die Universität Michigan, die LDAP 1993 entwickelte, den Ansatz, auf das weitverbreitete TCP/IP aufzusetzen und LDAP durch die reduzierte Komplexität benutzbarer zu machen.

Verzeichnisbaum

Jedes LDAP-Verzeichnis hat eine Baumstruktur, welche sehr schnelle Lesezugriffe ermöglicht. Das Wurzelelement, sowie jeder Knoten und die Blätter des Baumes (äußere Knoten) sind LDAP-Objekte und können ihrerseits Kinder besitzen. Jedes Objekt hat innerhalb des Verzeichnisbaums einen eindeutigen Namen, den „Distinguished Name“, kurz DN. Jedes dieser Objekte hat Attribute, welche im Verzeichnis eindeutige Namen besitzen. Die Werte, die die Attribute annehmen können, hängen von ihrem Typ (Syntax) ab. Der DN des Wurzelobjektes wird aus mindestens einem vorhandenen Wert mindestens eines seiner Attribute gebildet. Soll Beispielsweise das Wurzelelement die Organisation „Example“ repräsentieren, wäre ein möglicher DN „o=Example“. Dabei steht o für das Attribut „Name einer Organisation“. Möglich wäre für eine deutsche Niederlassung auch der DN „c=DE,o=Example“, wobei c dann für Attribut „Land“ steht. Der erste Teil des DNs aller weiteren Objekte wird dann analog zu dem des Wurzelknotens gebildet und dieser mit dem DN des Elternknotens verknüpft. Dieser erste Teil wird auch Relative Distinguished Name (RDN) genannt und muss innerhalb aller direkten Kindobjekte eindeutig sein (siehe Abbildung 8).

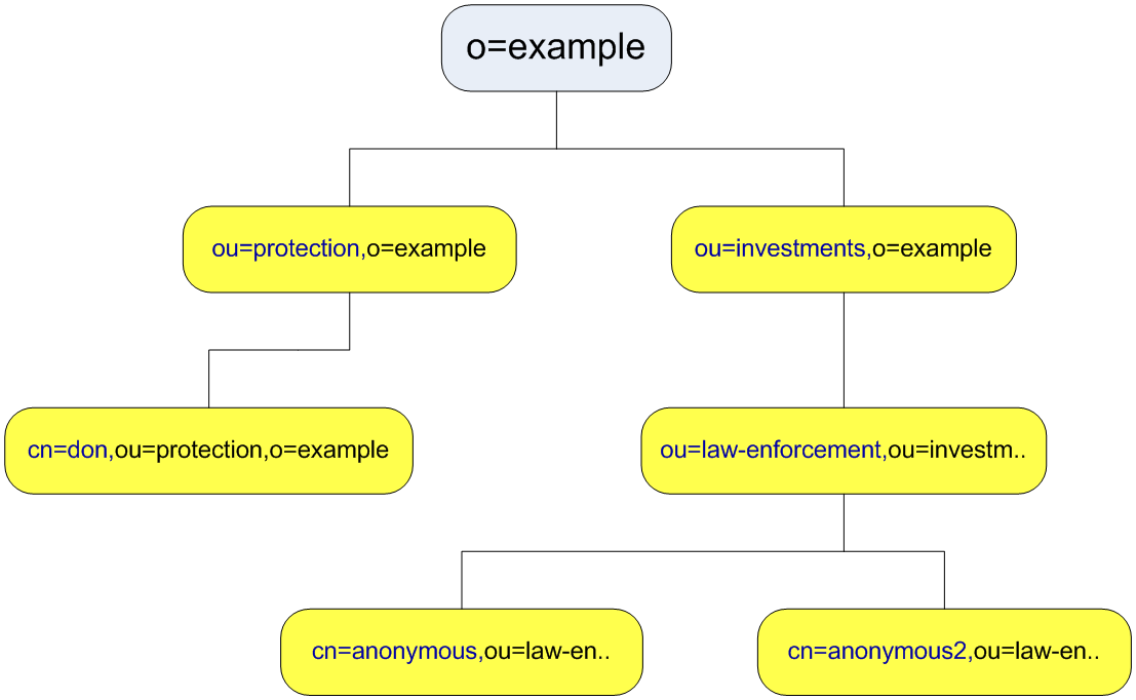

Abbildung 8: Verzeichnissbaum mit Distinguished Names

Objektklassen, Attribute, Syntax und LDAP-Schemata

Da LDAP einen objektorientierten Ansatz verfolgt, muss jedes Objekt die Instanz mindestens einer (strukturellen) Objektklasse sein. Die Klasse gibt vor, welche Attribute ein Objekt haben muss und welche es haben darf. Wie beim Klassenkonzept üblich, können Objekte erben und vererben. LDAP kennt 3 Typen von Klassen. Jedes Objekt muss mindestens zu einer „structural"- Klasse und kann zu mehreren „auxiliary" - Klassen gehören. „abstract"- Klassen können nicht direkt instantiiert werden, man muss sie erweitern.

Beispiel:

objectClass (2.5.6.6
NAME 'person'
DESC 'RFC2256: a person'
SUP top
STRUCTURAL
MUST (sn $ cn)
MAY (userPassword $ telephoneNumber $ seeAlso $ description))

Jedes Objekt besitzt Attribute. Diese werden definiert durch ihren Namen und den Datentyp des Wertebereichs. Zusätzlich lassen sich auf ihnen Vergleichsoperationen und Ordnungen definieren. Attributnamen sind im Verzeichnis eindeutig und können grundsätzlich mehr als einen Wert speichern.

Beispiel:

attributetype (2.5.4.13
NAME 'description'
DESC 'RFC2256: descriptive information'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1024})

Zum thematischen Zusammenfassen von Objektklassen-Definitionen gibt es das Konzept der LDAP-Schemata. Diese ermöglichen auch das freie Definieren von Objektklassen und Attributen. Schemata werden in Dateien gespeichert und in der Regel mit der Software ausgeliefert.
Um Klassen- und Attributdefinitionen eindeutig zu identifizieren, werden ObjectIdentifiers (OIDs) eingesetzt. Jeder Knoten und jedes Blatt wird durch eine eigene Nummer identifiziert. Die Eltern-Kind-Beziehung wird durch einen „.“ ausgedrückt. Die ATIS hat einen Subtree unterhalb der offiziell registrierten OID (1.3.6.1.4.1.87) des Instituts für Telematik der Universität Karlsruhe.

Da LDAP trotz seiner verminderten Funktionalität im Vergleich zu DAP ein sehr komplexes Protokoll ist, würde es den Rahmen dieser Studienarbeit sprengen, hier auf Details einzugehen. Dem interessierten Leser wird hiermit [RFC2251], welches LDAPv3 beschreibt, nahe gelegt, sowie [OLDP] für eine detaillierte Beschreibung von OpenLDAP, der für dieses Projekt eingesetzen open-source LDAP Software.

2.3 Web Services (WS)

Web Services sind Software-Anwendungen im Web. Sie sind durch einen Uniform Ressource Identifier (URI) eindeutig identifizierbar und bieten Anwendungsentwicklern die Möglichkeit, die eigene Software durch externe Anwendungen (die Web Services) zu erweitern, unabhängig von deren Plattform und Sprache.
Der Einsatz von WS basiert auf drei Standards: Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) und Universal Description, Discovery and Integration (UDDI). Web Services sind in Bezug auf Sicherheit noch nicht ausgereift, so dass eine gut durchdachte, sichere Infrastruktur von großer Bedeutung ist, wenn man WS-Schnittstellen anbieten will. Neue Entwicklungen im Gebiet Web Service Sicherheit bringt die Web Services Security (WSS) Technologie. [Ry03]

Simple Object Access Protocol (SOAP)

SOAP ist ein Standard zum Austausch von Nachrichten über das Internet. Ursprünglich wurde es entwickelt um Remote Procedure Calls (Aufruf von Prozeduren auf entfernten Systemen) über das HTTP-Protokoll zu realisieren. Für den Nachrichtenaustausch benutzt SOAP die XML-Technologie. [Ry03]

Web Service Description Language (WSDL)

Mit WSDL werden die Schnittstellen der Web Services beschrieben. Die Beschreibung umfasst das Protokoll, den Host, Portnummer sowie Operationen, Nachrichtenformate und Ausnahmen. Auch hier wird die XML-Technologie benutzt, nämlich in Form von XML-Schemata, die der Beschreibung eine Struktur verleihen. [Ry03]

Universal Description Discovery and Integration (UDDI)

UDDI ermöglicht das Suchen und Finden von Web Services. Es ist eine Art Verzeichnisdienst für Web Services. Anbieter von WS können diese per UDDI registrieren lassen und zusätzlich beschreiben, so dass Entwickler diese bei Bedarf finden. Es gibt 4 Typen von Einträgen: Business Entity, Business Service, Binding Template und Technology Model.
Eine Business Entity beschreibt ein Unternehmen, welches WS anbietet; ein Business Service ist eine Service-Klasse innerhalb eines Unternehmens. Ein WS, wie er von der WSDL beschrieben wird, wird durch die beiden verbleibenden Entitätstypen definiert. Wobei das Technology Model die abstrakte Ebene beschreibt und das Binding Template sich auf das tatsächliche Protokoll bezieht.[Ry03]
Für tiefere Einblicke in die WS-Technologie werden folgende Quellen empfohlen : [Ch06a], [Ch06b] sowie die darauf folgenden Artikel.

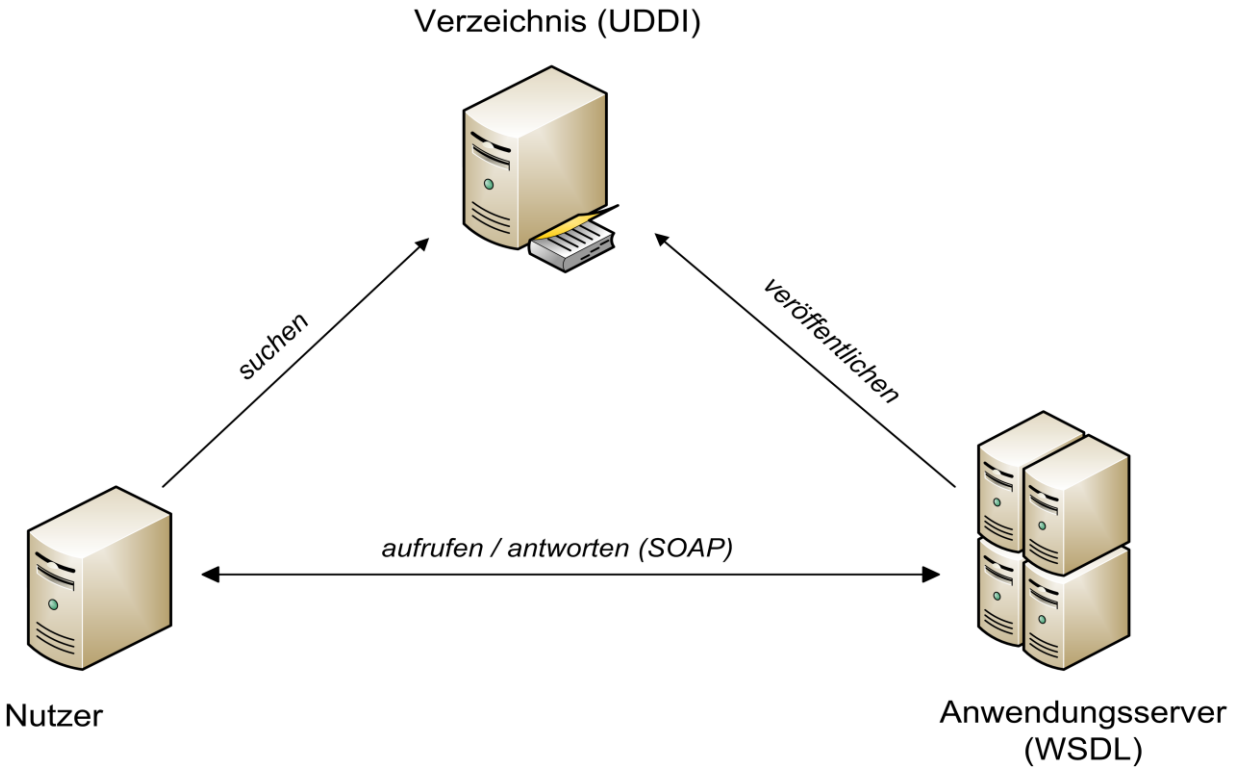

Abbildung 9: Web Services Übersicht

3 Analyse des bestehenden AM-Systems

In diesem Kapitel wird das bestehende System ausgehend von [HK01] beschrieben und im Anschluss bezüglich seiner Entwicklung, seinen Anforderungen und seiner Infrastruktur analysiert. Sinn der Analyse ist das Extrahieren der weiterhin benötigten Funktionalität und das Nutzen der aus Fehlern gewonnenen Erfahrung.

3.1 Initiale Systemstruktur

Abbildung 10 zeigt eine Übersicht des bestehenden Systems bei der Aufnahme des Betriebs, das sich im Wesentlichen aus zwei Teilen zusammensetzt:

- Die Managementanwendung UserADM für das Verwalten der Accounts
- Die Infrastruktur, um die Attribute der Accounts für die Dienstsysteme bereitzustellen (SQL-DB, LDAP-VZ und uadm2ldap.py-Skript)

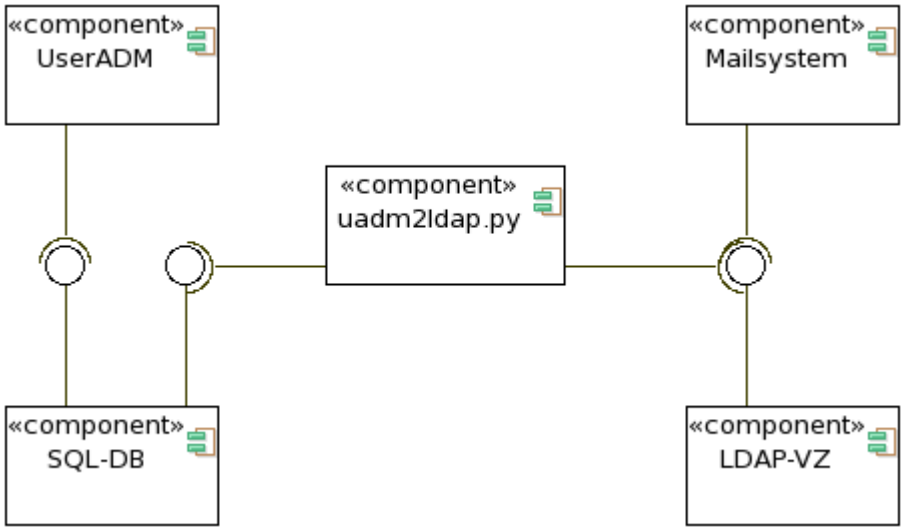

Abbildung 10: Initiale Systemstruktur mit Mailanbindung

Die Anwendung UserADM wurde als Teil einer Studienarbeit im April 2001 entworfen und implementiert [HK01]. Die Informationen zur Generierung der Nutzerkonten werden in einer SQL (Structured Query Language) – Datenbank in mehreren Tabellen gehalten und per Kommandozeile oder Web-Oberfläche bearbeitet. Auf der Anwendungsschicht wird die in Perl geschriebene Software durch die beiden Programme *useradm2.pl* (für die Web-Schnittstelle) und *useradm.pl* (für die ASCII-Schnittstelle) realisiert. Zwischen der Datenbasis und den Benutzerschnittstellen werden Perl-Module zur Umsetzung der Anwendung eingesetzt. Die Bereitstellung der Informationen (Attribute) der Nutzerkonten gegenüber den Dienstsystemen (z.B. Mailserver) wird durch ein LDAP-Verzeichnis realisiert. Die Komponente „uadm2ldap.py“ ist ein Python-Skript, welches dafür sorgt, dass die für das Mailsystem

relevanten Daten im LDAP im Falle einer Änderung in der SQL-Datenbank aktualisiert werden.
Aufgabe des Systems war zunächst das Bereitstellen eines dezentralen Managements von Mail-Accounts innerhalb der Fakultät, damit die Einrichtungen, unabhängig vom Betreiber, ihre Nutzer und die damit verbundenen Mail-Accounts einrichten und pflegen konnten. Das Mail-System selbst nutzt das LDAP-Verzeichnis für Anfragen, um eingehende Mails an die entsprechenden Postfächer auszuliefern oder sie im Falle einer unbekannten Adresse abzuweisen (Mail-Routing), aber nicht für die Authentifizierung der Nutzer, die ihre Mails lesen wollen. Diese Authentifizierung erfolgt bislang gegen die Datei „/etc/shadow“ auf dem Mail-Server (dort legen UNIX-Systeme die kodierten Passwörter der Nutzer ab). Das LDAP-Verzeichnis ist so strukturiert, dass es zusätzlich von den Mail-Klienten der Nutzer als Adressbuch genutzt werden kann.

Dezentrale Administration von Mail-Accounts

Aufgabe der Account-Management-Software (UserADM) ist es, den Administratoren der einzelnen Institute oder Einrichtungen die Möglichkeit zu geben, in ihrem Bereich eigenständig Mail-Accounts anzulegen und zu modifizieren. Ein solcher Account enthält neben Personendaten nur Attribute, die für das Mailsystem relevant sind.
Um eine dezentrale Administration zu ermöglichen, kennt die Software einen Super-Administrator, der fest im Source-Code verankert ist. Dieser hat das Recht, Benutzer und Accounts fakultätsweit zu verwalten. Zusätzlich kann er bestimmten Nutzern (den Administratoren der Institute) Administratorrechte für die jeweiligen Bereiche vergeben und auch wieder entziehen. Benutzer mit diesen Rechten sind dann in der Lage, sich an der Software anzumelden und können in ihrem Zuständigkeitsbereich eigenständig Benutzer und Accounts verwalten.
Bei der Vergabe der Benutzernamen ist der flache Namensraum zu beachten, d.h. fakultätsweit haben alle Nutzer E-Mail Adressen der Form „nachname@ira.uka.de“. Haben es nun zwei Personen denselben Nachnamen, kommt es zu einem Namens-konflikt, der durch Abweichen vom Namensschema gelöst wird.

3.2 Erweiterung der Systemstruktur

Durch neue Anforderungen wurde es notwendig, die oben beschriebene Struktur sukzessive zu erweitern. Diese Anforderungen und die damit verbundenen Erweiterungen werden im Folgenden beschrieben.
Die Authentifizierung der Studenten im Studentenpool der ATIS wurde bei Inbetriebnahme des Systems über einen Network Information Service (NIS), ein Verzeichnisdienst zum Verteilen von Konfigurationsdaten, realisiert. D.h. die Nutzerdaten wurden zwar per UserADM gepflegt, die Systeme fragten aber zum Authentifizieren gegen den NIS. Im Gegensatz zu LDAP ist NIS jedoch veraltet und unsicher (p.13)[Ma03], so dass bald ein zusätzlicher LDAP für die Authentifizierung der Studierenden an den Arbeitsplätzen im Studentenpool aufgesetzt wurde.

Die nächste Anforderung an das System war die Verwaltung des Druckguthabens der Studenten. Jeder Student, der einen Account für den Poolraum besitzt, bekommt jedes Semester ein Druckguthaben gutgeschrieben, das ihm erlaubt, eine gewisse Anzahl von

Seiten kostenlos im Poolraum zu drucken. Zusätzlich können sich Studenten Druckguthaben kaufen. Das Druckguthaben wurde zuerst in einer Berkeley Datenbank verwaltet, später wurden diese Informationen dann zur Konsolidierung der technischen Struktur ebenfalls in den studentischen LDAP verlagert. Der Übersicht wegen wurden die zwei LDAP-Verzeichnisse (eines für das Mailsystem, das andere für die Authentifizierung im Pool) schließlich zusammengeführt.

Zum Erstellen, Verändern und Löschen von Benutzern muss der Administrator mit der UserADM-Software direkt auf die SQL-Datenbank zugreifen. Von dort aus werden neue oder geänderte Datensätze (gekennzeichnet durch ein gesetztes sync-Flag) automatisiert in das LDAP-Verzeichnis übertragen. Zwar erfüllte das LDAP-Verzeichnis seinen Zweck sehr zufriedenstellend, aber es zeigten sich schnell Probleme, die aus dem Betrieb zweier Datenhaltungssysteme zur Verwaltung von Benutzerdaten resultierten. Weil bestimmte Attribute (wie das Passwort und das Druckguthaben) der Studenten bereits direkt im LDAP-Verzeichnis gespeichert wurden, wurden sie überschrieben, sobald das Synchronisationsskript den kompletten geänderten Datensatz von der SQL-Datenbank übertrug. Das Synchronisationsskript wurde daher so verändert, dass es die betroffenen Attribute vor der Aktualisierung sichert und danach wieder einfügt.

Seit die ATIS den Mitarbeitern und Studenten der Fakultät den Dienst Virtual Private Network (VPN) anbietet, werden die Zertifikate der Benutzer des VPN im LDAP abgelegt. Das hatte zur Folge, dass von nun an alle Mitarbeiter-Accounts vollständig in den LDAP aufgenommen werden mussten. Bisher waren nur die Studenten vollständig erfasst, von den Mitarbeitern waren lediglich die Attribute gespeichert, die der LDAP benötigte, um die Anfragen des Mailsystems zu befriedigen (Rückgabe des verantwortlichen Mailservers zu einer E-Mail Adresse und Auflösung der Aliase). Um aber das Passwort, das ein Mitarbeiter für das Mail-System hat, auch für VPN nutzen zu können, musste eine Änderung des Passworts für das Mailsystem dem LDAP gemeldet werden, wozu ein weiteres Synchronisationsskript geschrieben wurde. Eine unabhängige Änderung des VPN-Passworts wurde nicht vorgesehen Auch die IP-Nummern, welche die Benutzer vom VPN zugeteilt bekommen, werden im LDAP abgelegt.

Mit OpenLDAP [OLDP] existiert eine stabile, offene Implementierung des LDAP-Protokolls in der Version 3. Da sich LDAP von seiner inneren Struktur (Baumstruktur, objektorientiert) besser für unseren speziellen Einsatz eignet als SQL, wurde in Betracht gezogen, das System komplett auf LDAP umzustellen, statt SQL und LDAP zu verwenden. Zusätzlich würden bei dieser Umstellung sämtliche Synchronisationsmechanismen zwischen SQL und LDAP wegfallen, was die Administration erheblich vereinfachen würde.

3.2.1 Übersicht der aktuellen Anforderungen

Bei Inbetriebnahme des bestehenden Systems entsprach ein (Nutzer-)Account einem Mail-Account, der Mail-Attribute sowie Personenattribute beinhaltet. Durch die gestiegenen Anforderungen und die neuen Dienste musste der Inhalt eines Accounts entsprechend angepasst bzw. erweitert werden. Ein (Nutzer-)Account enthält nun

zusätzlich zu den Mail- und Personenattributen die Attribute für alle in Anspruch genommenen zentralen Dienste.
Im Folgenden werden die über die Zeit gewachsenen Anforderungen noch einmal anhand einer chronologisch geordneten Liste zusammengefasst:

- Dezentrales Management der Mail-Accounts (UserADM; SQL)
- Management der Benutzerdaten (UserADM; SQL)
- Mail-Routing (LDAP)
- Authentifizierung im Studentenpool (LDAP)
- Verwaltung des Druckguthabens der Studenten (LDAP)
- VPN für Mitarbeiter und Studenten (LDAP)

Um diese Anforderungen zu gewährleisten, müssen Dienste und Systeme von unserem AM-System bedient werden. Diese vom bestehenden System bedienten Dienste und Systeme werden im Folgenden stichpunktartig aufgelistet:

- Zentraler Mailserver (SMTP, POP3, IMAP)
- Studentischer Mailserver (SMTP, POP3, IMAP)
- Studentische Workstations (Authentifizierung, NameService, POSIX-Attribute)
- Studentische Druck-Accounts
- Studentische Server (SSH, WWW, FileServer (Home-Verzeichnisse))
- ATIS Workstations (Authentifizierung, NameService, POSIX-Attribute)
- ATIS Server (SSH, FileServer (Homes))
- ATIS VPN (+Dial-In)

3.2.2 Resultierende aktuelle Systemstruktur

In Abbildung 11 ist zu sehen, wie im bestehenden System die beteiligten Komponenten zusammenarbeiten. Daran wird deutlich, welche komplizierten Abhängigkeiten bestehen und welche Synchronisationsmechanismen wo eingesetzt werden. Betrachtet wird nur die Anbindung des Mailsystems und des VPN, da das Diagramm sonst zu unübersichtlich werden würde. Entstanden ist dieser sicherlich nicht befriedigende Zustand durch die notwendigen Erweiterungen (s.o.) und die Notwendigkeit, den kontinuierlichen Mail- und Poolbetrieb zu gewährleisten.

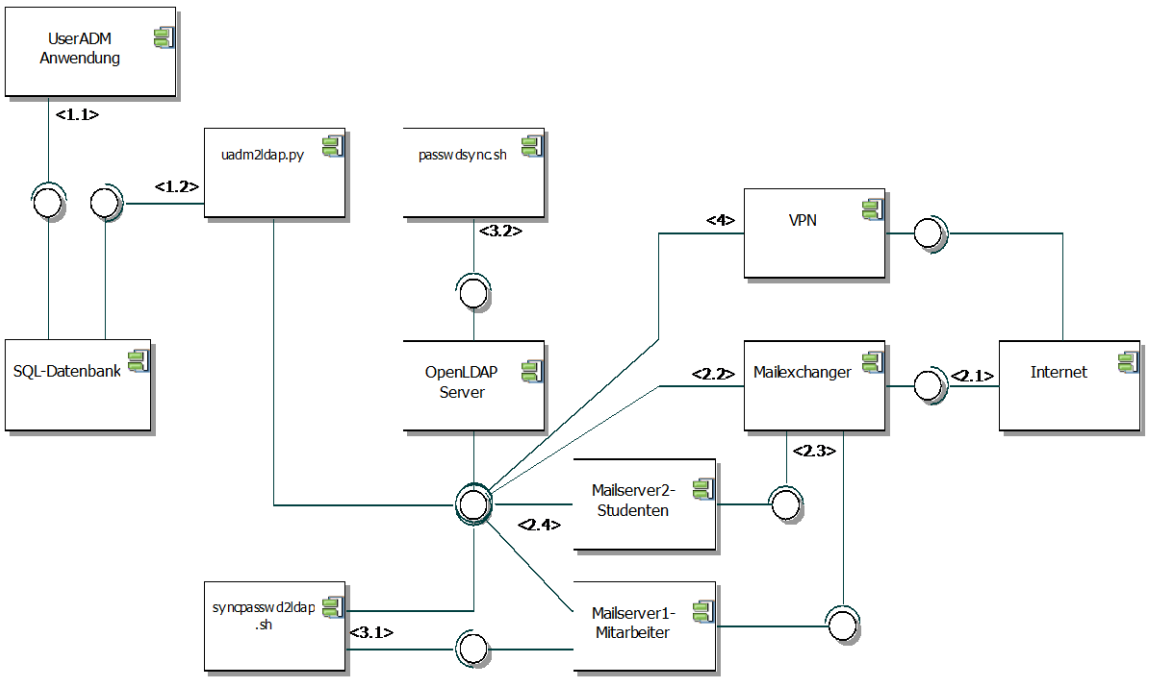

Abbildung 11: Gesamtstruktur mit Mail- und VPN-Anbindung

<1.1> : Mit der Oberfläche des UserADM (Web oder ASCII) kann direkt auf die SQL-Datenbank zugegriffen werden.

<1.2> : Per Cron-Job wird das Script *uadm2ldap.py* aufgerufen, das alle geänderten Datensätze (gekennzeichnet durch gesetztes sync-Flag) der SQL-Datenbank in den LDAP synchronisiert.

<2.1> : Vom Internet eingehende Mails landen in einem der beiden Mail-Exchanger MX1 oder MX2 (hier vereinfacht durch einen Mail-Exchanger dargestellt).

<2.2> : Kommt eine E-Mail an, prüft der Exchanger, ob der Empfänger bekannt ist, und falls ja, welcher Mailserver zuständig ist. Dazu stellt er eine Anfrage an das LDAP-Verzeichnis, welches die entsprechende Antwort zurückgibt.

<2.3> : Der Antwort des LDAP entsprechend wird die E-Mail an einen der beiden Mailserver weitergeleitet.

<3.1>, <3.2> : Das Shell-Skript *syncpasswd2ldap.sh* kopiert die Datei *passwd* des Mailservers auf den LDAP. Dort synchronisiert das Skript *passwdsync.sh* die Passwörter.

<4> : Der VPN Dienst nutzt den LDAP, um dort Zertifikate und IP-Nummern abzulegen.

3.3 Anwendungsfälle der Management-Anwendung

Im Folgenden werden die Anwendungsfälle der UserADM–Software erfasst und textuell beschrieben. Wir betrachten hierbei die Web-Schnittstelle, die textbasierte Variante besitzt den gleichen Funktionsumfang.
Nach der Anmeldung mit Benutzername und Passwort kommt man direkt in den Bereich „Startseite“, wo der Benutzer eine Übersicht über die Institute, für die er autorisiert ist, einsehen kann.
Der UserADM ist unterteilt in vier Bereiche, auf die mittels Menüleiste zugegriffen wird. Diese Bereiche sind im Einzelnen:

„Adressen suchen“:

‚Adressen suchen‘: Ermöglicht das Suchen von E-Mail Adressen anhand eines Suchkriteriums. Als Ergebnis werden alle Adressen geliefert, die einen Teil des Suchkriteriums enthalten, inkl. dem Typ der Adresse und ob diese im LDAP publiziert wurde.

„Benutzer suchen“:

‚Benutzer suchen‘: Ermöglicht das Suchen von Benutzern anhand der Suchkriterien Name und Vorname (es ist möglich nur einen Teil des gesuchten Namens als Kriterium zu verwenden). Zurückgeliefert werden die gefundenen Benutzer, die die Kriterien erfüllen, inklusive aller zugehörigen Accounts und aller Aliase. Auf der Ergebnisseite können sämtliche Benutzerdaten, Accounts und Aliase verwaltet werden.

„Unix-Accounts“:

‚Account suchen‘: Ermöglicht das Suchen von Accounts anhand der Kriterien Login, Verfallsdatum, UserID per Eingabefeld und der Kriterien Mailauslieferung und Klasse per Drop-Down Menü.

‚Account anlegen‘: Ermöglicht das Anlegen eines Unix-Accounts und dessen Zuordnung zu einem Benutzer. Beim Anlegen werden der Login-Name, das Institut, das Verfallsdatum und gegebenenfalls Bemerkungen benötigt. Zusätzlich können Häkchen für das Veröffentlichen im LDAP und in der Institutsliste gesetzt werden. Der Name des zugeordneten Benutzers muss ebenfalls angegeben werden. Vor dem Anlegen werden die Daten des gefundenen Benutzers nochmals aufgelistet sowie weitere Attribute des Accounts, die hier nochmal geändert werden können (Login, UID, GroupID, Verfallsdatum, Auslieferung, Institut, LDAP-Status, Klasse, Bemerkung, Institutsliste)

‚Benutzer anlegen‘: Ist im Fall ‚Account anlegen‘ der angegebene Benutzer nicht im System vorhanden, schlägt das System vor, diesen Benutzer inkl. dem Account (s.o.) anzulegen. Vor

dem Anlegen wird man nochmals aufgefordert, die Benutzer- und Account-Daten (s.o.) zu überprüfen und zu vervollständigen. Man muss bestätigen, den Benutzer auf die Benutzerordnung hingewiesen zu haben.

„Mail-Alias“:

‚Mail-Alias suchen‘: Ermöglicht das Suchen von Mail-Aliasen anhand eines Suchkriteriums. Liefert alle passenden Einträge zurück und ermöglicht das Löschen und Modifizieren der gefundenen Einträge.

‚Mail-Alias neu anlegen‘: Ermöglicht das Anlegen neuer Mail-Aliase und Mail-Forwards und die Zuordnung zu einem Benutzer. Ist dieser Benutzer nicht vorhanden, so ist es wie im Geschäftsprozess ‚Benutzer anlegen‘ möglich, den Benutzer direkt mit dem Alias anzulegen.

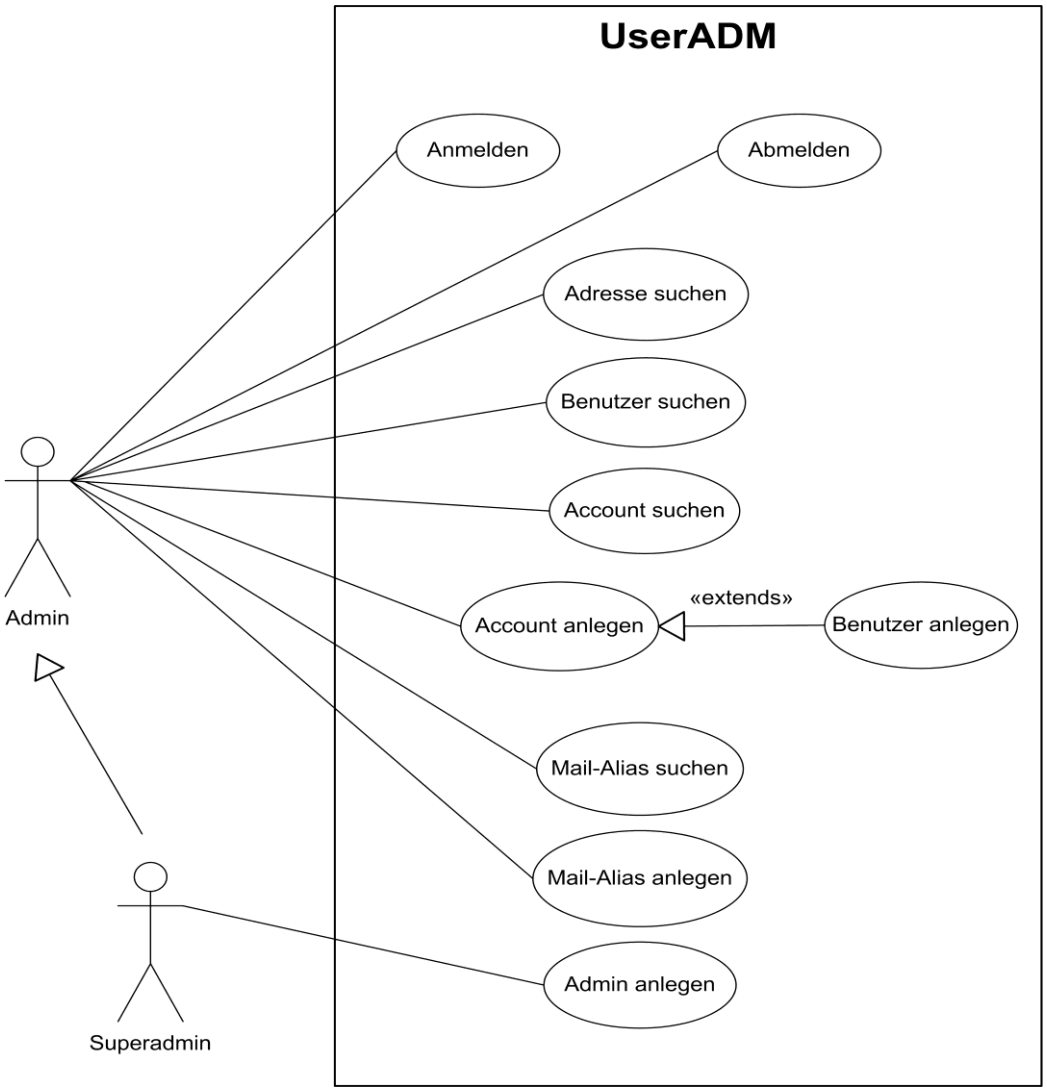

Abbildung 12: Anwendungsfalldiagramm des UserADM

3.4 Bewertung und Fazit „altes AM-System“

Ein großer Kritikpunkt am alten System sind die zwei eingesetzten heterogenen Datenhaltungssysteme SQL und LDAP. Um diese konsistent zu halten, ist eine umfangreiche Synchronisierung erforderlich. Dies hatte zusätzlich zur Folge, dass Änderungen in der Datenbank nicht sofort systemweit wirksam waren, sondern erst nach Ablauf aller Synchronisationsmechanismen.
Da der LDAP-Verzeichnisdienst fast alle Dienste bedient, liegt es nahe, die digitalen (Teil-)Identitäten der Benutzer und Accounts zukünftig direkt auf dem LDAP zu verwalten, statt den Umweg über die SQL-Datenbank zu gehen. Diese soll also komplett abgeschafft und die Benutzer direkt auf dem LDAP verwaltet werden.
Um diesen Zustand (zu viele heterogene Technologien, Synchronisationsmechanismen, aufwendiges Administrieren) beim neuen System zu vermeiden, muss von Beginn an auf eine möglichst einfache, modular erweiterbare Struktur geachtet werden.
Alle vom alten System bedienten Dienstsysteme müssen vom neuen System weiter bedient werden. Hier wird deutlich, welche Vorteile eine durchdachte Struktur bietet (vgl. hierzu „Abbildung 11 – Gesamtstruktur mit Mail- und VPN-Anbindung“ und „Abbildung 13 – Geplante neue Systemstruktur mit Mail- und VPN-Anbindung“).
Was die Funktionalität betrifft, soll das neue System alle Anwendungsfälle aus den Bereichen „Adressen suchen“, „Benutzer suchen“, „Unix-Accounts“ und „Mailalias/–forward“ übernehmen. Der aus vergangener Zeit stammende, fakultätsweit flache Namensraum des Mailsystems soll auf Wunsch vieler Institute abgeschafft werden. Das AM-System bis hinein in die Einrichtungen für deren lokale Dienste bereit zu stellen und zu nutzen, wäre ein weiterer großer Vorteil, da dann die Einrichtungen keine zusätzlichen, separaten Authentifizierungsmechanismen (z.B. NIS) pflegen müssten.

Übersicht der Mängel

- zwei heterogene Datenhaltungssysteme (SQL+LDAP)
- komplexe Synchronisationsmechanismen
- schwer erweiterbar, komplizierte Dienstanbindung
- keine Selbstbedienungsfunktion für Nutzer
- flacher Namensraum des Mailsystems
- berücksichtigt nur zentrale Dienste

4 Anforderungsanalyse des neuen AM-Systems

Dieses Kapitel bildet zusammen mit Kapitel 5 den Kern der Studienarbeit, die Anforderungsanalyse des neuen Systems. Bevor es im folgenden Kapitel an das Strukturieren und Modellieren der Anwendungsfälle geht, werden die Anforderungen herausgearbeitet, die bei der Modellierung berücksichtigt werden müssen. Dabei geht es sowohl um funktionale als auch um nicht-funktionale Anforderungen. Einen weiteren Punkt stellt die Anbindung an das KIM-IDM über externe Schnittstellen dar.

4.1 Funktionale Anforderungen

Die Funktionalität des Systems setzt sich aus den folgenden vier Punkten zusammen:

- Bereitstellen von Attributen für externe Dienste
- Dezentrale Verwaltung von Benutzerdaten – Accounts, Gruppen und Identitäten
- Verwaltung von Instituten und Einrichtungen
- Bereitstellen von Selbstbedienungsfunktionalität

4.1.1 Geplante neue Systemstruktur

Um die ersten drei der am Ende des letzten Kapitels aufgezählten Mängel (zwei heterogene Datenhaltungssysteme, komplexe Synchronisationsmechanismen und schwer erweiterbare Dienstanbindung) zu beseitigen, wird auf die SQL-DB verzichtet. Es bleibt also nur ein Datenhaltungssystem übrig, das LDAP-Verzeichnis, gegen das die nutzenden Dienste anfragen können und auf dem die Benutzerdaten der Anwendung verwaltet und abgelegt werden. Das hat zur Folge, dass sämtliche Synchronisationsmechanismen zwischen der ehemaligen SQL-DB und dem LDAP-Verzeichnis wegfallen und somit die Anbindung von Diensten sowie das Administrieren der Daten um ein Vielfaches einfacher wird. Für das neue System ergibt sich folgende, hier auf Mail und VPN beschränkte, Dienstanbindungsübersicht. Sie zeigt, wieviel einfacher das neue System im Vergleich zum Bestehenden mit der Dienstanbindung verfährt (zum Vergleich mit dem alten System siehe Abbildung 11):

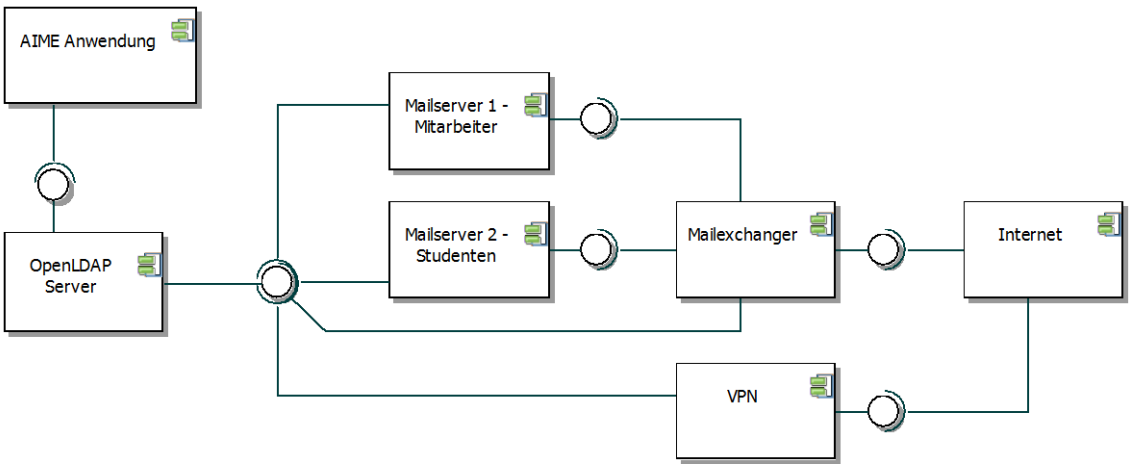

Abbildung 13: Geplante neue Systemstruktur mit Mail- und VPN-Anbindung

Die Dienste benötigen bestimmte Attribute eines Accounts, um einen Nutzer (seine Identität) zu authentifizieren und die für den weiteren Betrieb benötigten, identitätsabhängigen Werte zu erhalten. Das alte System berücksichtigte nur zentrale Dienste, das soll sich mit dem neuen System ändern. Dienste sollen auf verschiedenen Ebenen angeboten werden können, zum Beispiel nur innerhalb einer Einrichtung oder nur für bestimmte Nutzergruppen. Deswegen muss das neue AM-System verschiedene Accounttypen kennen. Das System wird zwischen Accounts, die für zentrale Dienste (z.B. VPN) fakultätsweit Attribute zur Verfügung stellen, und Accounts, die für einrichtungsspezifische Dienste (z.B. lokaler Fileserver) nur innerhalb einer Einrichtung Attribute zur Verfügung stellen, unterscheiden.
So wird sichergestellt, dass eine dezentrale Administration der einrichtungsbezogenen Accounts durch die Administratoren der jeweiligen Einrichtung weiterhin möglich ist.

Zentrale Accounts

Es gibt Dienste, die für die ganze Fakultät angeboten und von der ATIS verwaltet werden, z.B. E-Mail und VPN. Für diese Dienste ist es notwendig, Attribute einrichtungsübergreifend, also auf Fakultätsebene bereitzustellen. Dies wird durch einen zentralen Account realisiert, der als Container für Attribute einrichtungsübergreifender Dienste dient und Folgendes beinhaltet:

- Daten für das physische E-Mail Postfach (inkl. Login-Daten)
- Attribute einrichtungsübergreifender Dienste

Einrichtungsbezogene Accounts

Viele Dienste werden auf der Einrichtungsebene, d.h. nur innerhalb einer Einrichtung, angeboten. Diese auch auf Einrichtungsebene zu administrieren ist Grundlage für eine dezentrale Administration. Darum werden die Attribute für diese Dienste in einem einrichtungsbezogenen Account zusammengefasst. Dieser dient also analog zum zentralen Account als Container für Attribute einrichtungsbezogener Dienste. Dieser Accounttyp kann von den jeweiligen Administratoren in den entsprechenden Einrichtungen angelegt und administriert werden. Bevor für eine Person ein einrichtungsbezogener Account angelegt werden kann, sollte diese einen zentralen Account besitzen. Der zentrale Account ist aber aus technischer Sicht nicht notwendig für die Nutzung eines einrichtungsbezogenen Accounts. Einrichtungsbezogene Accounts werden beinhalten:

- Daten zur Authentifizierung / Autorisation gegen das AM-System
- Daten zur Authentifizierung / Autorisation gegen Workstations
- Attribute einrichtungsbezogener Dienste

Derzeit besteht kein Bedarf, Accounts auf Institutsebene anzubieten, da die Dienste entweder auf Einrichtungs- oder auf Fakultätsebene angeboten werden. Das heisst aber nicht, dass die Institutsebene unberücksichtigt bleibt, denn E-Mail Aliase sowie Accounts müssen innerhalb dieser Ebene auf Einzigartigkeit überprüft werden.

Gast- / Test-Accounts

Wie in Kapitel 1.2 schon angesprochen, werden die Identitätsdaten (im Folgenden Identitäten genannt) im neuen System über eine Schnittstelle vom KIM-IDM-System bereitgestellt und synchron gehalten. Für den Fall, dass ein Account angelegt werden muss, zu dem keine Identität bereitgestellt werden kann (z.B. für einen Gastdozenten, der durch die Verwaltung nicht erfasst wurde), behilft sich das System durch das Anlegen einer temporären Identität. Diese kann später durch eine vom KIM-IDM-System bereitgestellte Identität ersetzt werden - solange ein Account allerdings auf eine temporäre Identität zeigt, handelt es sich um einen Gast-Account.
Ein Test-Account wird von einem Administrator zu Testzwecken angelegt und verweist auf dessen Identität. Ein Administrator kann sich alle von ihm angelegten Test-Accounts anzeigen lassen, der Super-Administrator (Begriff wird im nächsten Abschnitt erläutert) kann sich alle Test-Accounts anzeigen lassen, unabhängig davon, ob er diese angelegt hat oder nicht.

Gruppen

Gruppen werden eingesetzt, um Account-Objekte zu strukturieren und so beispielsweise die Rechtevergabe der an das LDAP-Verzeichnis angebundenen Dienste zu vereinfachen. Ein Account wird bei seiner Generierung einer Gruppe zugewiesen, muss also mindestens Mitglied einer Gruppe sein. Diese ist seine primäre Gruppe. Zusätzlich zu dieser kann er weiteren (sekundären) Gruppen angehören.

4.1.2 Basisfunktionalität der Management-Anwendung

In erster Linie sollen durch die Anwendung die Benutzerdaten der Fakultät verwaltet werden. Dazu gehören neben Accounts und Identitäten auch die organisatorischen und logischen Einheiten, in denen diese sich befinden: Institute, Einrichtungen und Gruppen. Die Anwendung wird die bisher eingesetzte Software UserADM ablösen und muss daher den weiterhin benötigten Teil deren Funktionalität übernehmen. Zusätzlich soll sie neue Funktionalitäten, wie die Selbstbedienungs-Funktionen für Mitarbeiter und Studenten, und vor allem die Verwaltung der Accounts, nicht nur der E-Mail-Adressen, bereitstellen. Um die Selbstbedienungsfunktionen zu realisieren, wird neben den bisherigen Rollen „Administrator“ und „Super-Administrator“ die Rolle des „Users“ eingeführt. Jeder Nutzer des Systems ist obligatorisch ein „User“ und kann als dieser die neue Selbstbedienungsfunktionalität nutzen, d.h. er kann sein Passwort ändern, seine Account-Informationen einsehen etc. Soll er zusätzlich in der Lage sein, in einer (oder mehreren) Einrichtung(en) Accounts anzulegen und zu verwalten, muss er dazu die Rolle „Administrator“ für die jeweilige Einrichtung einnehmen. Diese Rolle kann nur der im LDAP festgelegte „Super-Administrator“ zuweisen. Das Einrichten und Verwalten von Einrichtungen und Instituten wird ausschließlich dem Super-Administrator erlaubt. Durch die Festlegung des Super-Administrators im LDAP gewinnt das neue System an Flexibilität gegenüber dem alten, in dem der Super-Administrator fest im Code „verdrahtet“ war. Bei der Beziehung zwischen den drei Rollen handelt sich um eine Generalisierungsbeziehung (nach UML), wie Abbildung 14

verdeutlicht. Die allgemeinste Rolle ist die des Users, gefolgt vom Administrator. Der speziellste Fall ist der des Super-Administrators.

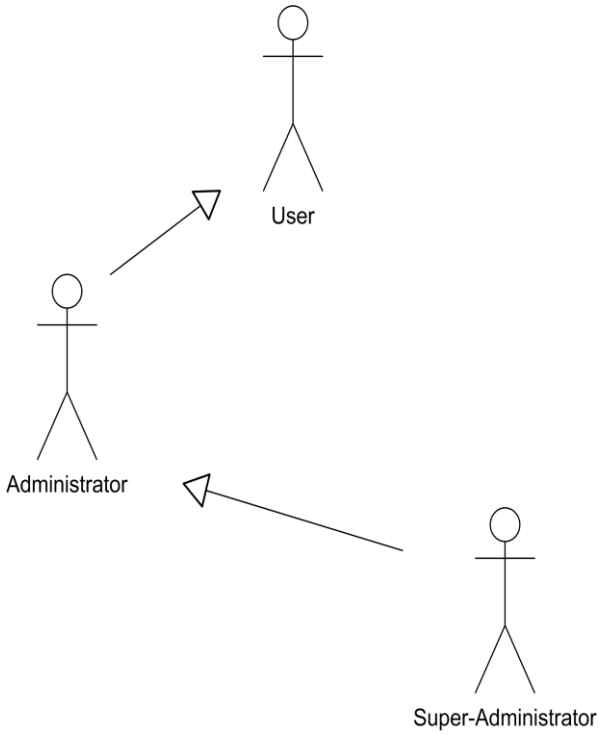

Abbildung 14: Generalisierungsbeziehung zwischen den Rollen

4.2 Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen an ein IT-System umfassen u.a. Aspekte wie die Architektur des Systems, die Datenhaltung oder die Sicherheit. Es geht um Fragen der Qualität und der inneren Struktur eines (Software-)Systems. Diese Art der Anforderungen wird nicht in der Modellierung der Anwendungsfälle berücksichtigt, denn diese repräsentieren die Funktionalität der Software. Aus diesem Grund ist es notwendig, die nicht-funktionalen Anforderungen hier ausführlich zu analysieren und festzuhalten. Allerdings ist es nicht einfach, sämtliche Anforderungen strikt nach funktional und nicht-funktional zu trennen, denn manche lassen sich nicht trivial in eines der beiden Felder einordnen.

4.2.1 Architektur des AM-Systems

Folgendes Komponentendiagramm zeigt die Architektur des neuen Systems, d.h. welche Komponenten über welche Schnittstellen miteinander kommunizieren. Die AM-Software wird als J2EE-Anwendung realisiert, die in einem Tomcat Server abläuft. Der Client greift mittels Web-Browser auf den Apache Server zu, der die eingehenden Anfragen über eine Proxy-Schnittstelle an die Anwendung weiterleitet. Die Anbindung an das KIM-IDM-System erfolgt per Web Service-Schnittstelle, über die das System die

Identitäten erhält (mehr zu dieser Schnittstelle in Abschnitt 4.3). Es hat sich gezeigt, dass die externen Dienste und die der ATIS sich sehr komfortabel an ein LDAP-Verzeichnis anbinden lassen. Darum fiel die Wahl für das Datenhaltungssystem auf ein LDAP Verzeichnis. Zum Einsatz kommt „OpenLDAP“ [OLDP] in der Version 2.3. Die Anbindung erfolgt über eine standardisierte LDAP Schnittstelle.

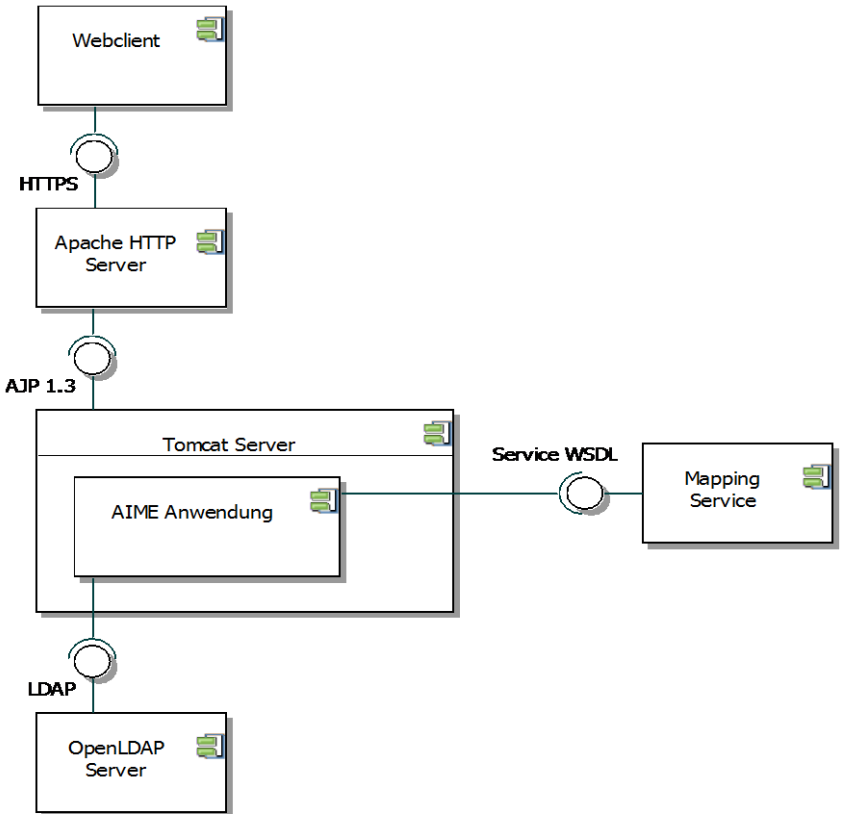

Abbildung 15: Architektur des neuen Systems

4.2.2 Datenhaltung

Da es sich bei dem AM-System um ein System zum Verwalten von Benutzerdaten handelt, ist der Datenhaltungsteil einer der komplexesten und wichtigsten Bereiche des Systems und somit auch dieser Studienarbeit. Zunächst werden die Anforderungen an die Datenhaltung stichpunktartig aufgeführt, bevor sie ausführlicher besprochen werden:

- Konsistente und persistente Speicherung aller für die Verwaltung und Dienstanbindung relevanten Daten
- Zentrales und direktes Bereitstellen der Daten für alle berechtigten Dienste
- Repräsentation der organisatorischen Gegebenheiten auf Ebene der Datenhaltung (innere Struktur der Datenhaltung)
- Eine möglichst flexible innere Struktur
- Einhalten der Datenschutzrichtlinien

Datenhaltungssysteme werden eingesetzt, um Daten konsistent und persistent abzulegen. Die nutzenden und berechtigten Dienste und Anwendung müssen in der Lage sein zu jedem Zeitpunkt auf die zentral gespeicherten Daten zuzugreifen. Dies sollte ohne den Einsatz von selbstgeschriebenen Softwarebrücken geschehen, d.h. die Anbindung der Dienste an die Datenhaltung sollte möglichst direkt und standardbasiert erfolgen. Das erhöht die Modularität und sorgt für eine geringere Komplexität bei der Wartung des Systems. Aus diesem Grund sei hier noch einmal erwähnt, dass sich sehr viele Dienste unproblematisch an ein LDAP-Verzeichnis anbinden lassen, da sie die dafür benötigten nativen Schnittstellen mitbringen.
Eine Repräsentation der organisatorischen Gegebenheiten auf Ebene der Datenhaltung ist aus mehreren Gründen von Vorteil. Es ermöglicht einerseits eine verteilte Administration und andererseits das Abbilden der DNS-Struktur der Fakultät auf die Datenhaltung, was wiederum das Aufbrechen des flachen Namesraums der Fakultät ermöglicht.
Damit Änderungen der organisatorischen Gegebenheiten (z.B. das Gründen eines neuen Instituts) schnell und sauber in das System übernommen werden können, sollte die innere Struktur der Datenhaltung möglichst flexibel sein. Das Einhalten der Datenschutzrichtlinien ist obligatorisch und wird durch die Entkoppelung von Identitäten und Accounts vereinfacht. Accounts können somit bearbeitet werden, ohne dass der Administrator mit nicht benötigten, identitätsabhängigen Attributen konfrontiert wird.

Erfassen der organisatorischen Gegebenheiten

Auf Ebene der Datenhaltung müssen die Geschäftsobjekte (Identitäten, Accounts, Gruppen, Einrichtungen und Institute) den organisatorischen Gegebenheiten entsprechend repräsentiert werden. Hierzu ist es notwendig die Geschäftsobjekte und ihre Beziehungen zueinander formal zu erfassen wie in Abbildung 16 in einem Klassendiagramm geschehen. Es zeigt die mengenmäßigen Restriktionen bei der Erstellung von (Geschäfts-)Objekten im LDAP-Verzeichnis. So kann beispielsweise ein Account nicht existieren, ohne dass ihm eine Identität zugeordnet wird, was die „ 1 “ an der Kante zwischen dem Identitäts- und dem Account-Objekt auf Seite der Identität festlegt. Auf der gegenüberliegenden Seite legt die Beschriftung „ 0..* “ fest, dass einer Identität keine bis beliebig viele Accounts zugewiesen werden können. Für genauere Erklärungen zu Kardinalitäten von Assoziationen sei erneut auf [BR+06] hingewiesen.

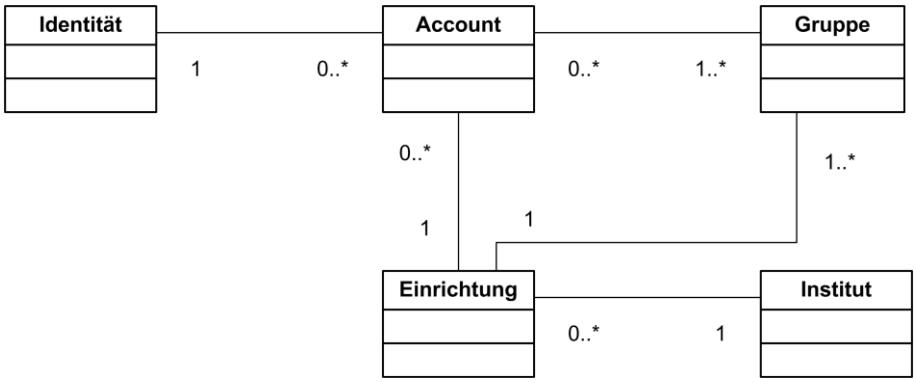

Abbildung 16: Beziehungen zwischen den Geschäftsobjekten

Verzeichnisstruktur und Beschreibung der Geschäftsobjekte

Die in Abbildung 16 gezeigten Beziehungen zwischen den Geschäftsobjekten werden im LDAP-Verzeichnis durch die Verzeichnisstruktur umgesetzt. Diese spiegelt einerseits die Struktur der Datensätze wieder und hält sich andererseits an die durch Abbildung 16 repräsentierten Restriktionen.

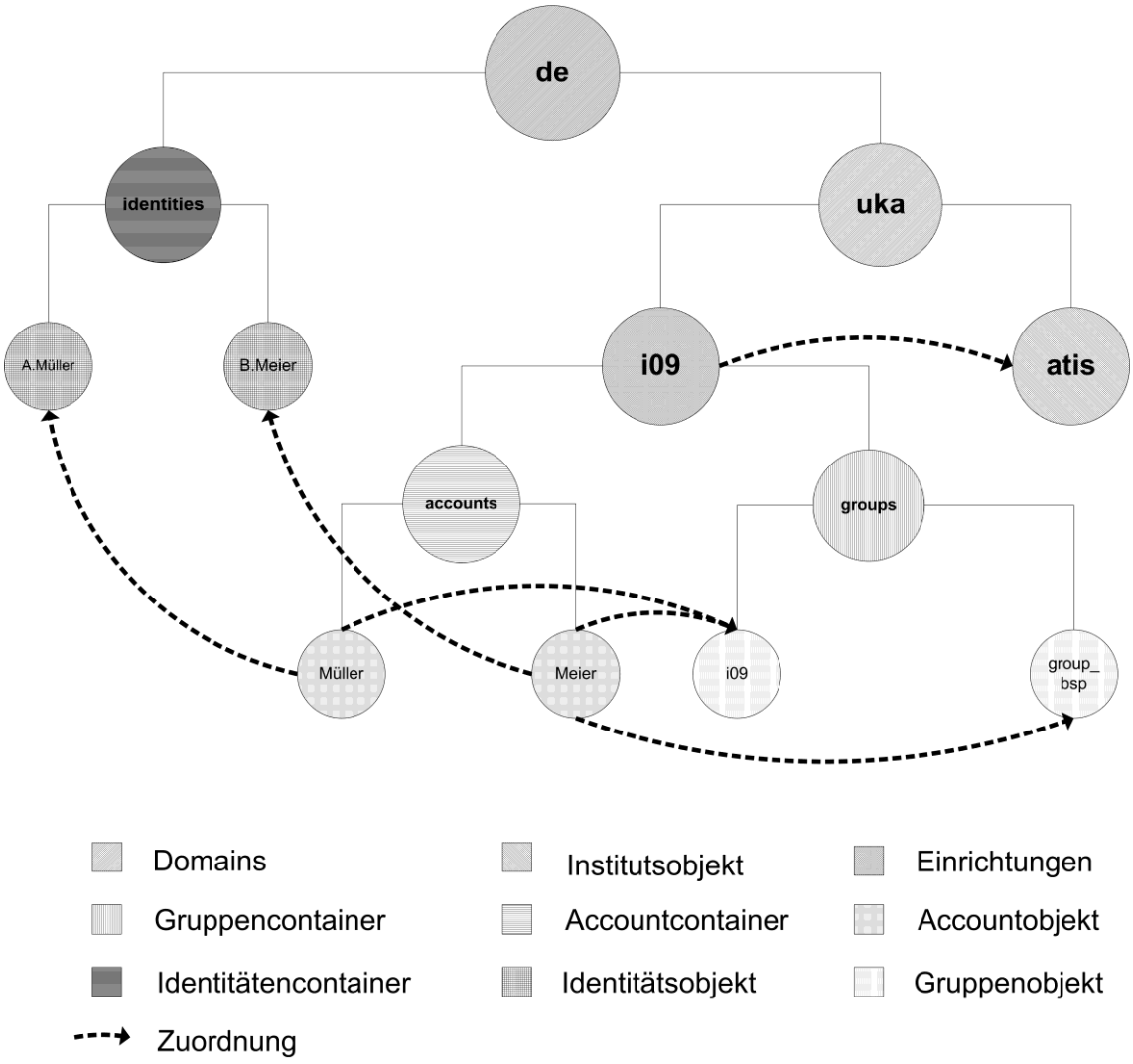

Abbildung 17: Verzeichnisstruktur

Account-Objekte

Jedes Account-Objekt besitzt die gleichen, in Tabelle 1 gelisteten Basisattribute. Wie in Kapitel 4.1 schon erwähnt wurde, kennt das System verschiedene Account-Typen. Bei einem Account kann es sich sowohl um ein (zentrales) Postfach mit E-Mail Adresse handeln, als auch um einen vollwertigen E-Mail- und POSIX-Account (POSIX-Accounts enthalten alle für die Anmeldung an Workstations relevanten Attribute). Für welchen Zweck der Account benutzt wird, zeigt sich durch die in einem Account-

Objekt gesetzten Attribute. Zusätzlich wird zwischen einrichtungsbezogenen und zentralen Accounts unterschieden, wie es bereits bei den funktionalen Anforderungen erläutert wurde. Diese unterscheiden sich anhand ihrer zusätzlichen Attribute, die in Tabelle 2 und Tabelle 3 aufgelistet werden, und ihrer Position im LDAP-Verzeichnisbaum.

Atrributname	Object Classes	Beschreibung
cn, commonName	posixAccount	Name (hier Name der Person)
gidNumber	posixAccount	ID der primären Gruppe
homeDirectory	posixAccount	Pfad zum Benutzerprofil
uid, userId	posixAccount	Benutzername des Accounts
uidNumber	posixAccount	Numerische ID des Accounts
loginShell	posixAccount	Pfad zur Loginshell des Accounts
userPassword	posixAccount	Passwort zur Authentifizierung
gecos	posixAccount	POSIX-spezifische Beschreibung
atisMail, atisLocalPart	atisMailProfile	Local Part der Mailadresse
atisMailForward	atisMailProfile	E-Mail-Adresse, zu der alle empfangenen Mails weitergeleitet werden
atisMailHost	atisMailProfile	FQDN des Postfachservers
atisMailMsgStore	atisMailProfile	Pfad zur physischen Mailbox auf dem MailHost
atisMailShell	atisMailProfile	Pfad zur Loginshell auf dem MailHost
atisVacation	atisMailProfile	Text einer Benachrichtigungsmail im Urlaubsfall
atisCreationTimestamp	atisMaintenance	Zeitpunkt der Accounterstellung
atisCreator	atisMaintenance	Referenz zum Identitätsobjekt des Accounterstellers
atisModifyTimestamp	atisMaintenance	Zeitpunkt der letzten Änderung am Account
atisModifier	atisMaintenance	Referenz zum Identitätsobjekt des Verantwortlichen für letzte Änderung am Account
atisExpiration	atisMaintenance, atisTestObject	Zeitpunkt, bis zu dem der Account bestehen bleiben soll
atisOwner	atisMaintenance, atisTestObject	Referenz zum Identitätsobjekt des Account-Inhabers
atisRemarks	atisMaintenance, atisTestObject	Verwaltungstechnische Bemerkungen
atisInstituteRoster	atisMaintenance	Entscheidung, ob der Account in die Instituts(mailing)liste aufgenommen werden soll
atisDisabled	atisMaintenance	Sperrt den Account (Authentifizierung!), falls 'TRUE'
atisPublicDirectory	atisMaintenance	Entscheidung, ob der Account in den fakultätsweiten Adressserver aufgenommen werden soll

Tabelle 1: Standard-Attribute eines Account-Objekts

Atrributname	Object Classes	Beschreibung
atisVpnIp	atisVpnProfile	IP-Adresse des VPN-Interfaces

Tabelle 2: Zusatz-Attribut zentraler Accounts

Atrributname	Object Classes	Beschreibung
atisDeposit	atisAccounting	Gekauftes Guthaben in Anzahl S/W-Seiten
atisSemesterDeposit	atisAccounting	Freies Semesterguthaben in Anzahl S/W-Seiten

Tabelle 3: Zusatz-Attribute einrichtungsbezogener Accounts

Gruppenobjekte

Gruppen werden eingesetzt, um Accounts zu strukturieren. Eine Gruppe speichert dazu die Identifikatoren (UIDs) aller ihr zugewiesenen Accounts. Jeder Account muss mindestens einer (primären) Gruppe zugewiesen sein. Diese Zuweisung geschieht im Account-Objekt über das Attribut „gidNumber“. Ist ein Account zusätzlich zu seiner primären Gruppe Mitglied einer oder mehrerer weiterer Gruppen, ist seine „uid“ im Gruppen-Attribut „memberUid“ gespeichert.

Atrributname	Object Classes	Beschreibung
cn, commonName	posixGroup	Name (hier Name der Account-Gruppe)
gidNumber	posixGroup	ID der Gruppe
memberUid	posixGroup	Loginname der Accounts, die Mitglied in der Gruppe sind
atisCreationTimestamp	atisMaintenance	Zeitpunkt der Gruppenerstellung
atisCreator	atisMaintenance	Referenz zum Identitätsobjekt des Gruppenerstellers
atisModifyTimestamp	atisMaintenance	Zeitpunkt der letzten Änderung an der Gruppe
atisModifier	atisMaintenance	Referenz zum Identitätsobjekt des Verantwortlichen für letzte Änderung an der Gruppe
atisExpiration	atisMaintenance, atisTestObject	Zeitpunkt, bis zu dem die Gruppe bestehen bleiben soll
atisRemarks	atisMaintenance, atisTestObject	Verwaltungstechnische Bemerkungen

Tabelle 4: Attribute eines Gruppenobjekts

Identitätsobjekte

Identitätsobjekte verhalten sich ähnlich wie Gruppenobjekte. Jedes Account-Objekt muss genau einer Identität zugewiesen sein. Ein Identitätsobjekt kann jedoch auf

mehrere Accounts zeigen. Es wird zwischen regulären Identitäten (Datensatz wird über die KIM-IDM-Schnittstelle bereitgestellt) und temporären Identitäten (KIM-IDM stellt keinen Datensatz bereit) unterschieden. Temporäre Identitäten werden über das Attribut „atisTemporaryIdentity“ gekennzeichnet.

Atrributname	Object Classes	Beschreibung
cn, commonName	atisIdentity	Enthält den vollständigen Namen der Person
gn, givenName	atisIdentity	Enthält den Vornamen der Person
sn, surname	atisIdentity	Enthält den Nachnamen einer Person
atisId, atisIdentifier	atisIdentity	Enthält den ATIS-Internen, eindeutigen Identifier der Person
telephoneNumber	atisIdentity	Enthält eine Telefonnummer, unter der die Person erreichbar ist
fax	atisIdentity	Enthält eine Faxnummer, unter der die Person erreichbar ist
title	atisIdentity	Enthält die Titel "Professor" und/oder "Doktor", wenn zutreffend
homePhone	atisIdentity	Enthält eine Festnetznummer, unter der die Person erreichbar ist
jpegPhoto	atisIdentity	Enthält ein JPEG-kodiertes Foto der Person
mobile	atisIdentity	Enthält eine Handynummer, unter der die Person erreichbar ist
userCertificate	atisIdentity	Enthält ein Zertifikat der Person, ausgestellt von der CA der Uni Karlsruhe
atisExternalId	atisIdentity	Enthält den eindeutigen Identifier für die Person im KIM-IDM
atisExternalExpired	atisIdentity	Wird auf 'TRUE' gesetzt, falls die Identität nichtmehr im KIM-IDM besteht
atisTemporaryIdentity	atisIdentity	Wird auf 'TRUE' gesetzt, falls die Identität im KIM-IDM (noch) nicht vorhanden ist
atisEnrolledStudent	atisStudent	Speichert den Immatrikulationsstatus an der Uni Karlsruhe
atisItStudent	atisStudent	Ist 'TRUE', falls die Person an der Fakultät für Informatik immatrikuliert ist
atisMatriculationNumber	atisStudent	Kann die Matrikelnummer an der Uni Karlsruhe speichern, falls dies nötig wird
atisCreationTimestamp	atisMaintenance	Zeitpunkt der Identitätserstellung
atisCreator	atisMaintenance	Referenz zum Identitätsobjekt des Identitätserstellers
atisModifyTimestamp	atisMaintenance	Zeitpunkt der letzten Änderung an der Identität
atisModifier	atisMaintenance	Referenz zum Identitätsobjekt des Verantwortlichen für letzte Änderung an der Identität

Tabelle 5: Attribute eines Identitätsobjekts

Einrichtungsobjekte

Einrichtungsobjekte repräsentieren Einrichtungen. Eine Einrichtung muss einem Institut angehören, deswegen bedarf es mindestens eines Institutsobjekts, auf das verwiesen wird. Einrichtungsobjekte beinhalten einen Container für Account-Objekte.

Atrributname	Object Classes	Beschreibung
ou	atisDepartment	Name der Einrichtung

Tabelle 6: Attribute eines Einrichtungsobjekts

Institutsobjekte

Institutsobjekte repräsentieren Institute. Institute können sich weiter in Einrichtungen unterteilen, in diesem Fall wird von der Einrichtung auf das Institut verwiesen. Im LDAP liegen Einrichtungen und Institute auf derselben Ebene. Account-Objekte, bzw. deren Container, können direkt unter einem Institutsobjekt liegen, falls sie keiner Einrichtung angehören.

Atrributname	Object Classes	Beschreibung
dc	atisInstitute	Name/Kürzel des Instituts

Tabelle 7: Attribute eines Institutsobjekts

4.2.3 Passwort-Policy

Um die Sicherheit zu erhöhen, werden die Passwörter aller Accounts (zentral und einrichtungsbezogen) voneinander unabhängig sein. So kann der Nutzer für jeden Account ein eigenes Passwort benutzen. Ändert ein Nutzer sein einrichtungsbezogenes Passwort, so hat er die Möglichkeit, sein zentrales Passwort gleich mit zu ändern. Weicht das zentrale Passwort vor der Änderung vom Einrichtungsbezogenen ab, so muss er vor der Änderung das alte zentrale Passwort erneut eingeben, um sich zu authentifizieren. Um die Sicherheit weiter zu erhöhen, ist eine starke Passwort-Policy vorgesehen.

4.2.4 Erweiterbarkeit und Modularität

Aus Erfahrung mit dem alten System ist bekannt, dass sich die Anforderungen an Software-Systeme sehr häufig ändern bzw. dass die Anforderungen im Laufe der Zeit stark zunehmen (siehe Abschnitt 2.2 – Die Historie des alten Systems). Dies führt schnell zu einem unüberschaubaren und schwer zu administrierenden System. Ein Grund dafür ist unter anderem die schnelle Entwicklung neuer Technologien im Software- sowie im Hardwarebereich. Daher ist einer der grundlegenden Design-Aspekte des neuen Systems Modularität. Es sollte möglich sein, das System durch vordefinierte Mechanismen in sämtlichen Bereichen ohne Veränderung des

ursprünglichen Codes zu erweitern ohne den Betrieb zu unterbrechen. Nur so kann darauf abgezielt werden, das neue System besser zu erweitern und zu pflegen als das Bestehende. Um dies zu ermöglichen, müssen Architektur und Schnittstellen des neuen Systems von Anfang an durchdacht und robust definiert werden.

4.2.5 Konsistenz und Aktualität der Benutzerdaten

Eine hohe Konsistenz und Aktualität der verwalteten Benutzerdaten ist eine weitere Anforderung, die nicht nur prinzipiell wünschenswert ist, sondern durch die implizit die Sicherheit erhöht werden würde; so sind z.B. weiterbestehende Accounts von ehemaligen Mitarbeitern ein nicht zu unterschätzendes Sicherheitsrisiko.
Da das System mit einem einzelnen Datenhaltungssystem arbeitet, gibt es nur die Schnittstelle zum KIM-IDM-System beachten, die das System mit Identitäten versorgt. Hier sollten Änderungen der Identitätsdaten unserem System möglichst schnell mitgeteilt werden.

4.3 Externe Schnittstellen

Web-Service Schnittstelle zum KIM-IDM-System

Alle verwaltungstechnisch relevanten Attribute einer Identität werden durch die Universitätsverwaltung erfasst. Eine für das IDM relevante Teilmenge dieser Identitätsdaten wird von der Verwaltung an das KIM-IDM-System weitergegeben. Das KIM-IDM-System stellt dann dem AM-System der Fakultät für Informatik eine (aus datenschutzrechtlichen Gründen) erneut eingeschränkte Teilmenge dieser Identitätsdaten über eine Web Service-Schnittstelle zur Verfügung. Somit können Identitätsdaten (Identitäts-Objekte mit vereinbarter Attributmenge) beim Anlegen eines neuen Accounts ohne Neuerfassen übernommen werden. Änderungen der Attributwerte (z.B. Immatrikulationsstatus) können dem AM-System so zeitnah und ohne manuelles Eingreifen eines Administrators zur Verfügung gestellt werden. Soll ein Account angelegt werden, zu dem durch die Verwaltung, also diese Schnittstelle, keine Identität geliefert werden kann, so muss eine temporäre Identität geschaffen werden (siehe Abschnitt Gast-/Test-Accounts). Zum jetzigen Zeitpunkt lassen sich aufgrund fehlender Spezifikationen der Schnittstelle noch keine weiteren Angaben dazu machen. Dies liegt noch im Aufgabenbereich des KIM-Projekts und ist kein Fokus dieser Arbeit.

5 Modellierung der Anwendungsfälle

Nach der Analyse der Anforderungen werden nun die Anwendungsfälle der neuen Software erfasst, strukturiert und in einzelnen Fällen modelliert.

5.1 Übersicht

Es folgt eine tabellarische Übersicht aller (grobgranularen) Anwendungsfälle (siehe Kapitel 2.1 – UML), gegliedert nach den nutzenden Akteuren. Akteure sind Objekte, die mit dem System bzw. der Software über Anwendungsfälle interagieren. In unserem Fall sind die Akteure die in Kapitel 4.1.2 eingeführten Nutzer-Rollen: User, Administrator, Super-Administrator; Akteure müssen aber nicht menschlich sein, so könnten z.B. andere Systeme auch einen Akteur darstellen.
Die Akteure stehen untereinander in einer Generalisierungsbeziehung, d.h. jeder Nutzer des Systems ist automatisch in der Rolle des „User“ und kann die für den User gedachten Anwendungsfälle benutzen (siehe Kapitel 4.1.4 – Abbildung 14).

5.1.1 Anwendungsfälle für User

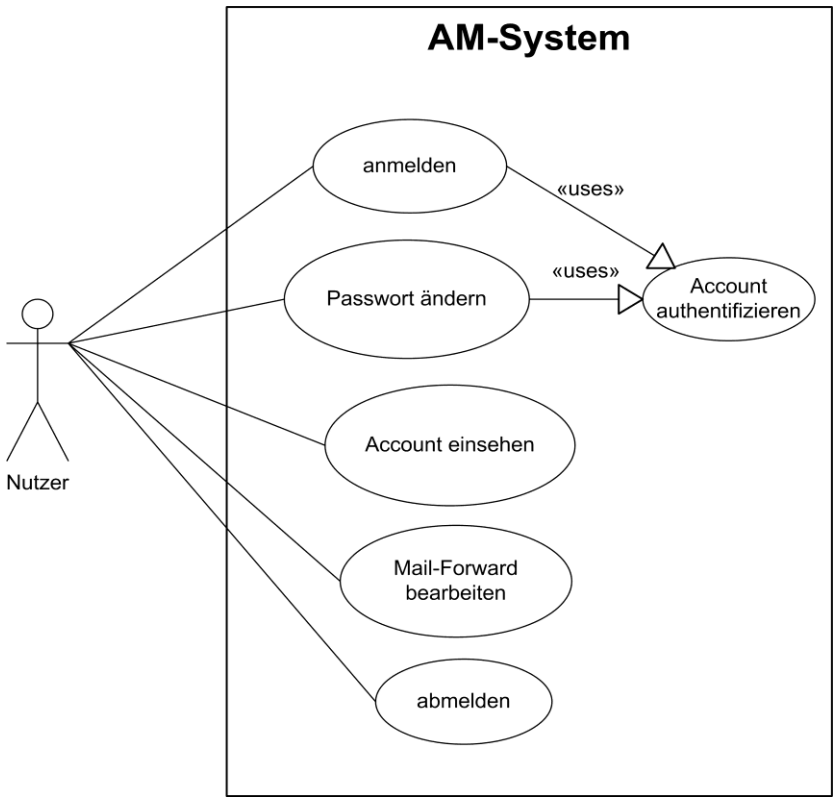

Abbildung 18: Anwendungsfalldiagramm "User“

Name	benutzt	erweitert	Beschreibung
Anmelden	Account authentifizieren	-	Am System anmelden
Account einsehen	-	-	Statusinformationen des eigenen Accounts ansehen
Passwort ändern	Account authentifizieren	-	Das eigene Passwort ändern
Mailforward einrichten	-	-	Eine Mail-Weiterleitung einrichten
Abmelden	-	-	Vom System abmelden
Account authentifizieren	-	-	Überprüft die Login-Daten auf Gültigkeit

Tabelle 8: Anwendungsfälle "User"

5.1.2 Anwendungsfälle für Administratoren

Bestimmten Personen wird aufgrund ihrer Anmeldung die Rolle Administrator für eine oder mehrere Einrichtung(en) (z.B. eines Forschungsbereichs) zugewiesen. Administratoren besitzen folgende Anwendungsfälle (in ihrem Bereich):

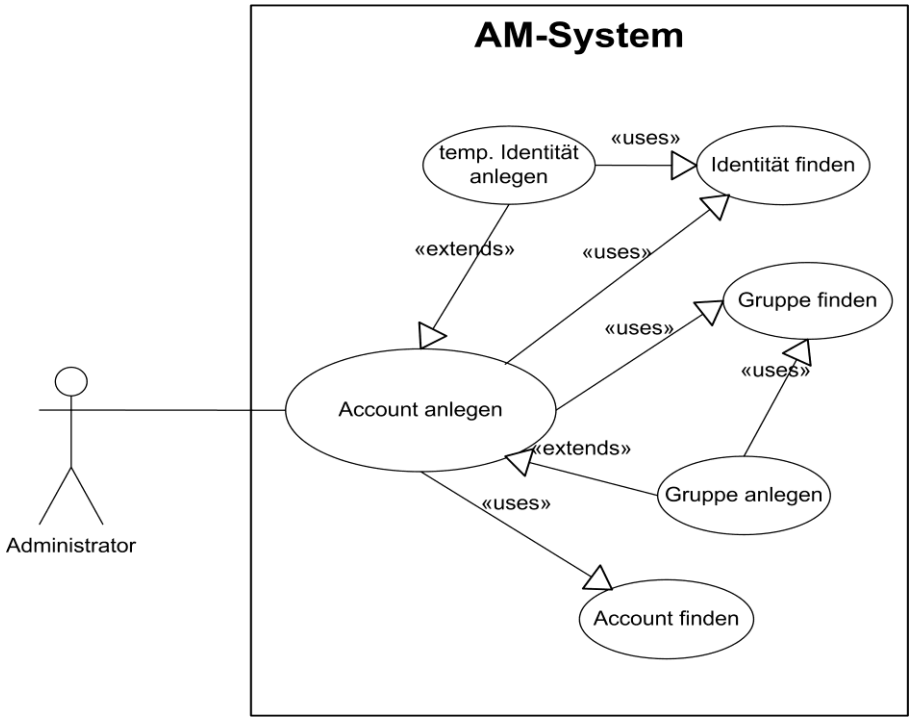

Abbildung 19: Anwendungsfalldiagramm „Administrator – Account anlegen“

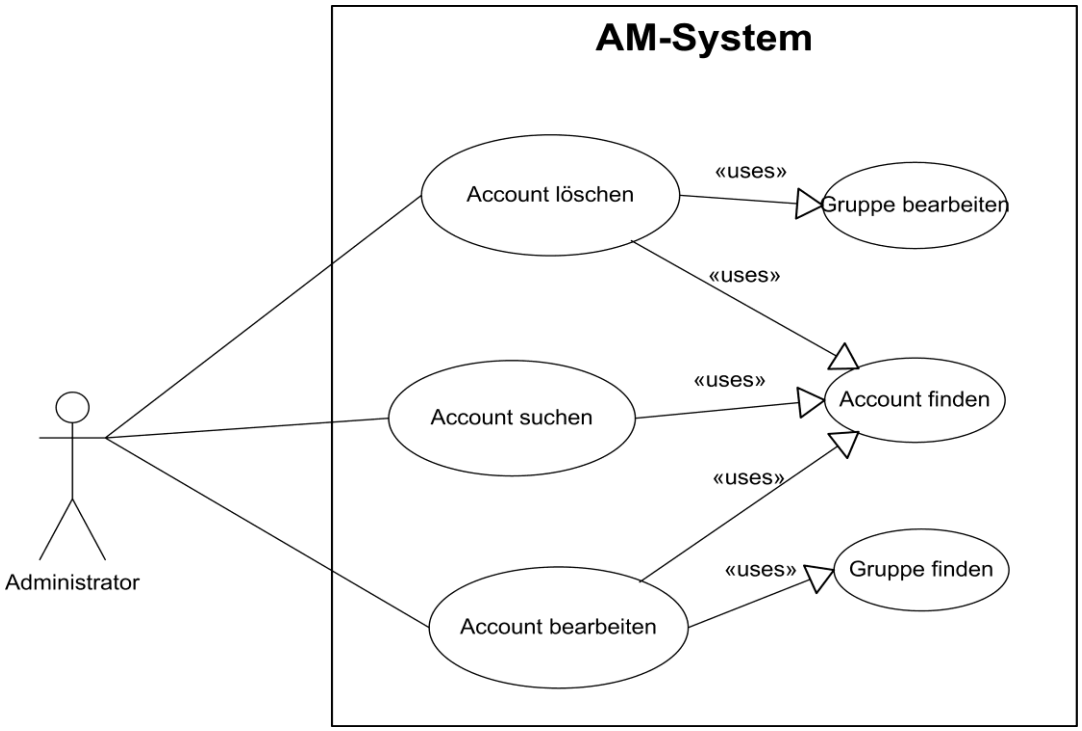

Abbildung 20: Anwendungsfalldiagramm „Administrator – Accounts“

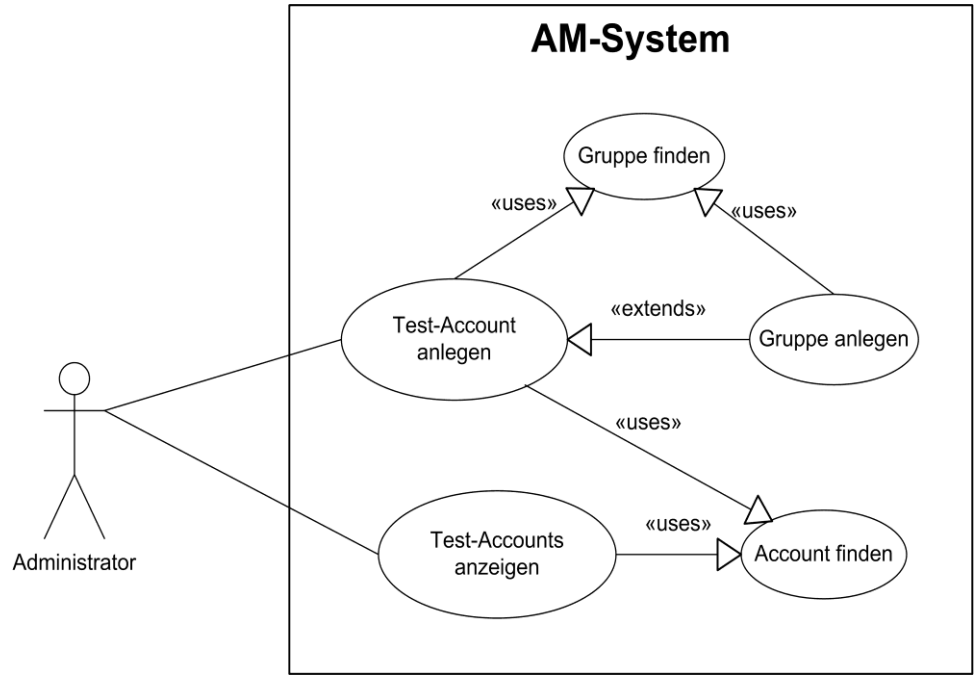

Abbildung 21: Anwendungsfälle "Administrator - Test-Accounts"

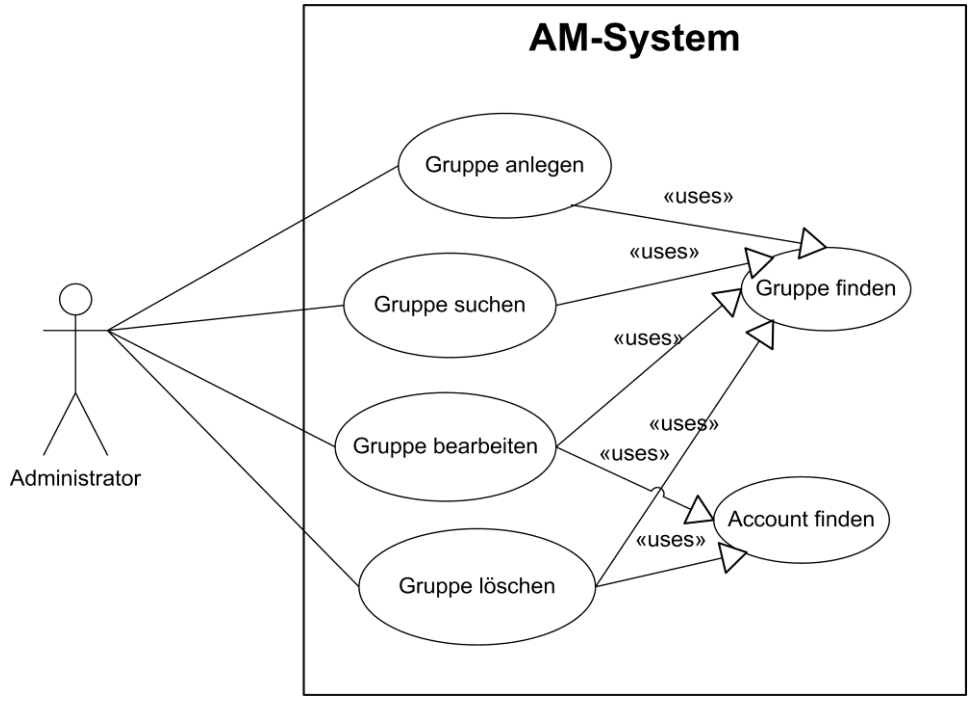

Abbildung 19: Anwendungsfalldiagramm „Administrator - Gruppen"

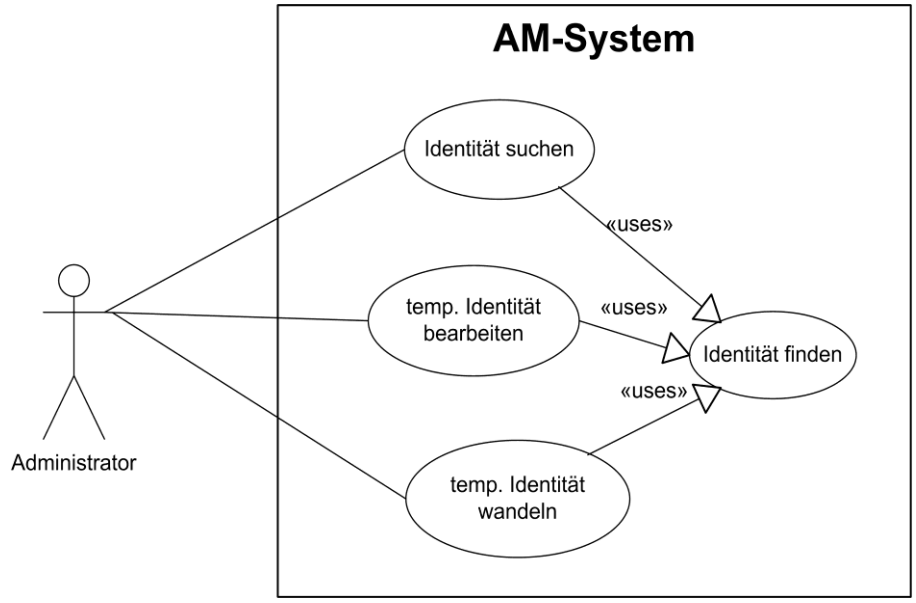

Abbildung 20: Anwendungsfalldiagramm "Administrator - Identitäten"

Name	benutzt	erweitert	Beschreibung
Identität suchen	Identität finden	-	Zeigt die Attribute einer Identität an
Identität finden	-	-	Identität anhand eines Begriffs suchen
Temp. Identität anlegen	Identität finden	Account anlegen	Legt eine temp. Identität an.
Temp. Identitat bearbeiten	Identität finden	-	Verändert die Attribute einer Identität
Temp. Identitäten wandeln	Identität finden	-	Wandelt eine temp. Identität bei Verfügbarkeit in eine normale Identität.
Account anlegen	Identität finden Account finden Gruppe finden	-	Legt einen einrichtungsbezogenen Account an
Account suchen	Account finden	-	Zeigt die Attribute eines Accounts an
Account bearbeiten	Account finden Gruppe finden	-	Verändert die Attribute eines Accounts
Account löschen	Account finden Gruppe bearbeiten	-	Löscht einen Account
Account finden	-	-	Sucht einen Account anhand eines Textes
Test-Account anlegen	Account finden Gruppe finden	-	Legt einen einrichtungsbezogenen Test-Account an
Test-Accounts anzeigen	Account finden	-	Zeigt alle vom Admin angelegten Test-Acounts an
Gruppe anlegen	Gruppe finden	Account anlegen Test-Account anlegen	Legt eine Gruppe an
Gruppe suchen	Gruppe finden	-	Eine Gruppe suchen
Gruppe bearbeiten	Gruppe finden Account bearbeiten	-	Verändert die Attribute einer Gruppe
Gruppe löschen	Gruppe finden Account finden	-	Löscht eine Gruppe
Gruppe finden	-	-	Findet eine Gruppe anhand eines Textes

Tabelle 9: Anwendungsfälle "Administratoren"

5.1.3 Anwendungsfälle für den Super-Administrator (Einrichtungen)

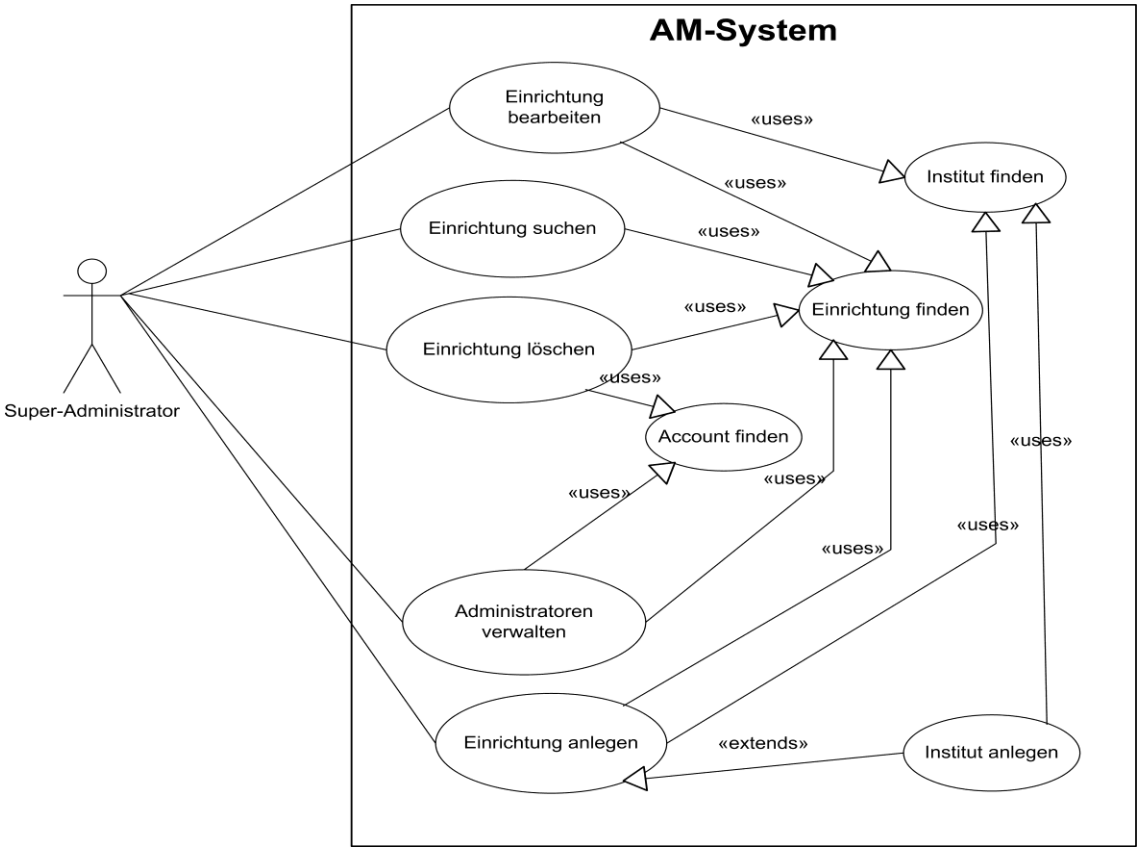

Abbildung 21: Anwendungsfalldiagramm "Super-Administrator – Einrichtungen“

Name	benutzt	erweitert	Beschreibung
Einrichtung anlegen	Einrichtung finden Institut suchen	-	Legt eine neue Einrichtung an
Einrichtung bearbeiten	Institut suchen Einrichtung suchen	-	Attribute einer Einrichtung ändern
Einrichtung suchen	Einrichtung finden	-	Attribute einer Einrichtung einsehen
Einrichtung finden	-	-	Sucht eine Einrichtung anhand eines Textes
Einrichtung löschen	Einrichtung finden Account finden	-	Ein Einrichtung löschen
Administratoren verwalten	Einrichtung finden Account finden		Administratoren ernennen und löschen
Institut anlegen	Institut finden	Einrichtung anlegen	Legt ein neues Institut an

Tabelle 10: Anwendungsfälle "Super-Administrator - Einrichtungen"

5.1.4 Anwendungsfälle für den Super-Administrator (Institute)

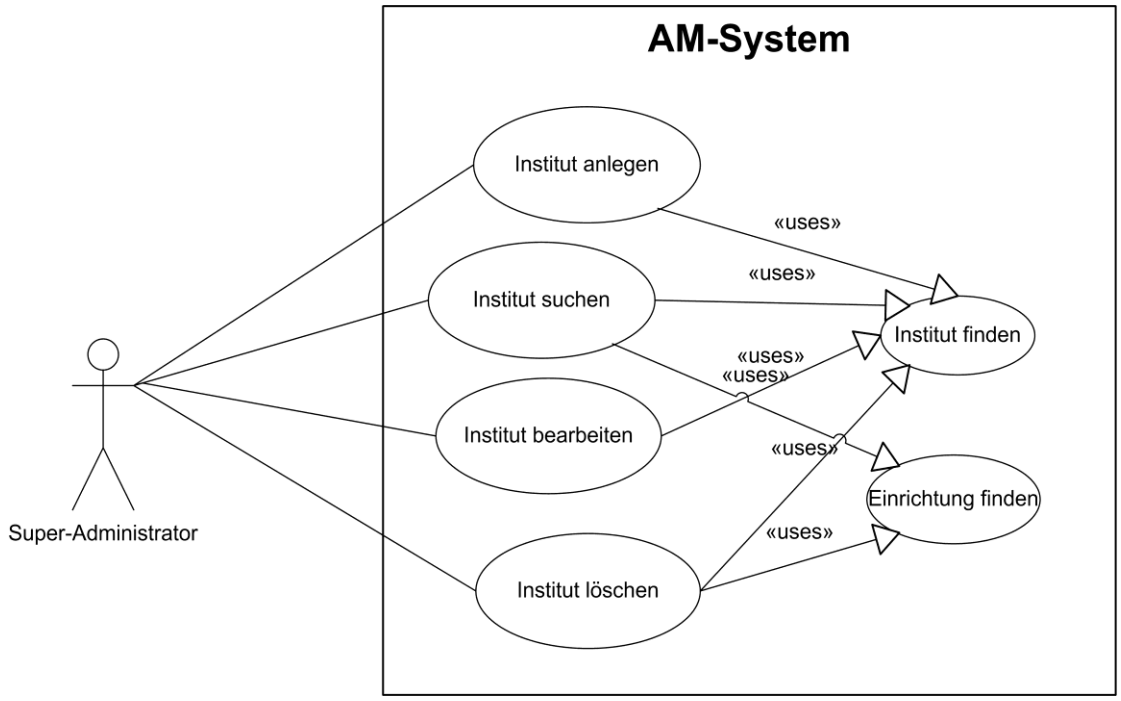

Abbildung 22: Anwendungsfalldiagramm "Super-Administrator - Institute"

Name	benutzt	erweitert	Beschreibung
Institut anlegen	Institut finden	-	Legt ein neues Institut an
Institut bearbeiten	Institut finden	-	Attribute eines Instituts ändern
Institut suchen	Institut finden	-	Attribute eines Instituts einsehen
Institut löschen	Institut finden	-	Ein Institut löschen
Institut finden	-	-	Sucht ein Institut anhand eines Textes

Tabelle 11: Anwendungsfälle "Super-Administrator - Institute"

5.1.5 Anwendungsfälle für den Super-Administrator (sonstige)

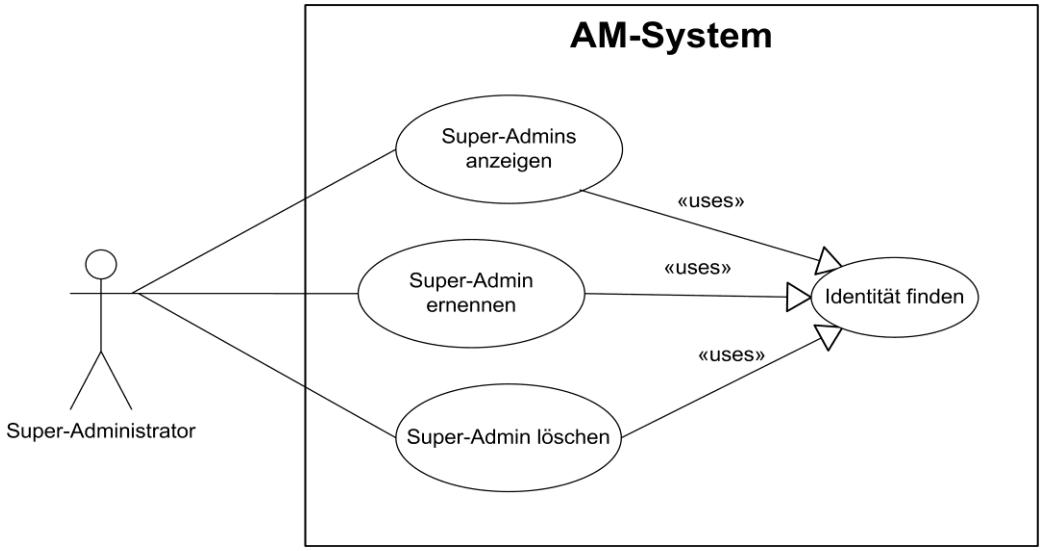

Abbildung 23: Anwendungsfalldiagramm "Super-Administrator – sonstige“

5.2 Der Anwendungsfall „Account anlegen“ (Administrator)

In diesem Abschnitt wird exemplarisch für die Anwendungsfälle der Administratoren der Fall „Account anlegen“ ausführlich beschrieben. Die Wahl fiel auf den Fall „Account anlegen“, da dieser der wichtigste und umfassendste aller Administratoren-Anwendungsfälle ist. Es ist zu beachten, dass es sich hier um einen einrichtungsbezogenen Account handelt.

5.2.1 Tabellarische Beschreibung

Name	Account anlegen
Ziel	Ein Administrator kann innerhalb seines Zuständigkeitsbereiches (Einrichtung) selbstständig Accounts anlegen.
Vorbedingung	Nutzer ist eingeloggt und hat Administratorrechte
Nachbed.(Erfolg)	Neuer Account wurde angelegt
Nachbed. (Fehler)	Neuer Account wurde nicht angelegt
Akteure	Administrator
Auslöser	Auswahl des Menüpunkts „Account anlegen“
Beschreibung	Der Nutzer muss alle für den Account relevanten Daten in ein Formular eingeben. Im nächsten Schritt werden diese überprüft, und bei korrekten Eingaben kann der Account im LDAP angelegt werden.

Tabelle 12: Anwendungsfall "Account anlegen

5.2.2 Aktivitätsdiagramm

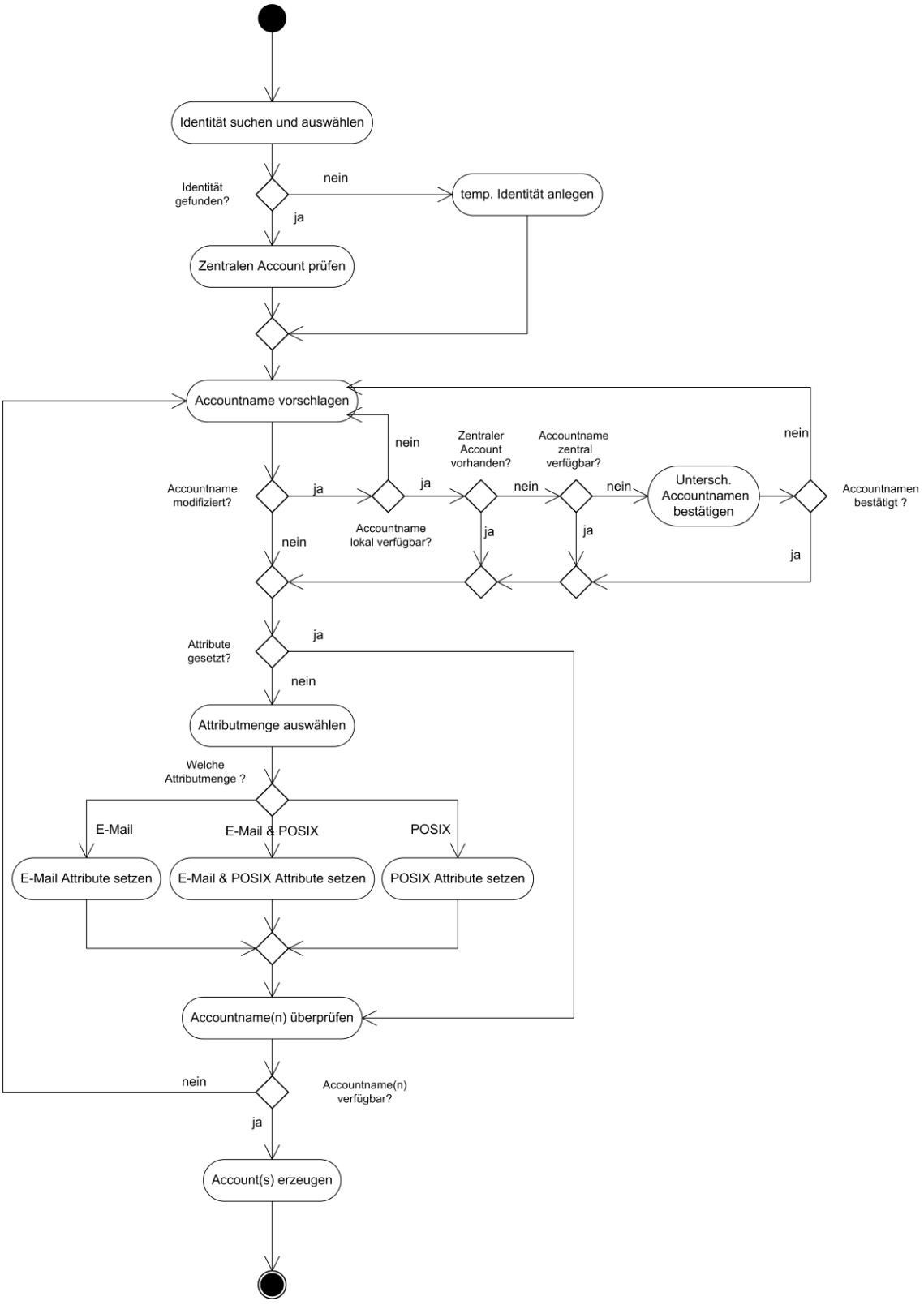

Abbildung 24: Aktivitätsdiagramm „Account anlegen“

5.2.3 Textuelle Beschreibung

Nach Anwählen des Menüpunkts „Account anlegen“ im entsprechenden Menü der Anwendungssoftware befindet sich der Administrator im Startpunkt des Aktivitätsdiagrammes und gelangt sofort zur Aktivität „Identität suchen und auswählen“. Hier kann nach einem Begriff gesucht werden, zu dem alle verfügbaren Identitäten in einem Drop-Down-Menü angezeigt werden. Die gesuchte Identität kann ausgewählt werden, und es wird überprüft, ob bereits ein zentraler Account zu der gewählten Identität existiert. Ist dies nicht der Fall, wird der zentrale Account automatisiert angelegt, sobald ein Account-Name ermittelt wurde. Existiert ein zentraler Account, so merkt sich die Software den Account-Namen. Sollte die gesuchte Identität im Drop-Down-Menü nicht zu finden sein, kann über einen Schalter eine temporäre Identität angelegt werden. Da zu neuangelegten temporären Identitäten kein zentraler Account existieren kann, wird die Prüfung nach dem zentralen Account übersprungen.
In der nächsten Aktivität „Account-Name vorschlagen“ wird das System dem Administrator einen Namen für den anzulegenden Account vorschlagen. Ist ein zentraler Account vorhanden, so wird der Account-Name des zentralen Accounts vorgeschlagen, wenn dieser auch lokal noch frei ist.
Im einfachsten Fall ist der Administrator bzw. der Nutzer damit einverstanden, d.h. er hätte für den neu angelegten lokalen Account und seinen zentralen Account denselben Account-Namen. Ist er mit dem Vorschlag nicht einverstanden, so kann er diesen modifizieren. Der modifizierte Name wird dann im nächsten Schritt auf Verfügbarkeit im lokalen Bereich geprüft. Ist er bereits vergeben, so wird das System einen neuen Vorschlag machen, der wiederum modifiziert werden kann. Ist der modifizierte Name lokal verfügbar, erfolgt eine Prüfung auf zentraler Ebene, falls kein zentraler Account existiert. Ist er auch auf zentraler Ebene verfügbar, so gelangt er direkt zur nächsten Aktion „Attributmenge festlegen“. Ist der Name zentral bereits vergeben, so bekommt der Administrator einen Hinweis, dass sich bei Beibehaltung des modifizierten Namens der zentrale und der lokale Account namentlich unterscheiden werden. Der Administrator kann diesen Hinweis bestätigen und gelangt dadurch zur nächsten Aktivität. Ist er damit nicht einverstanden gelangt er zurück zur Aktion „Account-Name vorschlagen“.
Die Aktivität „Attributmengen festlegen“ erlaubt dem Administrator auszuwählen, welche Attribute des neuen Account-Objekts bei dessen Erzeugung gesetzt werden und dadurch zu bestimmen, wozu der Account genutzt werden kann. Zur Auswahl stehen drei Möglichkeiten:

- E-Mail Attribute
- POSIX Attribute
- E-Mail & POSIX Attribute

Nach Wahl der Attributmenge prüft das System nochmals die Verfügbarkeit der/des gewählten Account-Namens bevor das Account-Objekt im LDAP angelegt wird. Ist der Name bereits vergeben, gelangt der Nutzer zurück zur Aktion „Account-Name vorschlagen“, die er erneut durchlaufen muss. Das Setzen der Attribute wird übersprungen.

5.2.4 Problem- und Fragestellungen

Bei den Diskussionen zum Anwendungsfall „Account anlegen“ waren die primären Fragestellungen, in welchem Verhältnis der zentrale und der lokale Account zueinander stehen sollen und wie man bei der Wahl der Account-Namen vorgehen sollte.
Es wurde entschieden, dass zentrale Accounts nur angelegt werden können, wenn die Software diesen Prozess anstößt, falls dieser aus technischer Sicht notwendig wird (siehe Abb.26). Mit dem zentralen Account ist es nicht möglich, sich am System anzumelden und das Passwort zu ändern oder Account-Informationen einzusehen. Dies geht nur über einen lokalen Account. Meldet man sich also mit einem lokalen Account am System an, so bekommt man zusätzlich zu den Informationen zum lokalen Account alle relevanten Informationen zum zentralen Account. Ändert man das Passwort seines lokalen Accounts, so gibt es die Option, über das Setzen eines Hakens, das zentrale Passwort mitzuändern. Allerdings muss sichergestellt werden, dass im ersten angelegten lokalen Account alle Attribute zum Anmelden an das System gesetzt sind. Es sollte sich daher um einen POSIX oder einen kombinierten (E-Mail & POSIX) Account handeln.

Die Vergabe der Account-Namen wurde bis zur letzten Besprechung diskutiert. Hier gab es zwei Standpunkte: Möglichkeit 1 sah für den zentralen Account eine Kombination aus Buchstaben und Zahlen und dadurch einen eindeutigen Account-Namen vor, während Möglichkeit 2 den Nachnamen bzw. einen dem Nachnamen möglichst ähnlichen Account-Name präferierte. Der Vorteil dieser zweiten Möglichkeit wäre ein einfach zu merkender Account-Name und in vielen Fällen eine Gleichheit von lokalem und zentralem Account-Name. Der Nachteil wäre ein relativ hohes Konfliktpotential auf zentraler Ebene. Dieses Konfliktpotential entsteht bei Möglichkeit 1 nicht, da hier durch die Eindeutigkeit des Namens Konfliktfreiheit erreicht wird. Der Nachteil dieser Möglichkeit ist, dass dieser Name nicht so einfach zu merken ist und durch die Abweichung der zentralen und lokalen Account-Namen das Bewusstsein für den zentralen Account geschaffen wird. Letztendlich wurde entschieden, dass Möglichkeit 2 implementiert wird.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

Wie diese Studienarbeit gezeigt hat, ist die Entwicklung eines Systems zum Verwalten von Accounts und den zugehörigen Nutzerdaten eine komplexe Aufgabe.
Nachdem der Leser in das Themengebiet eingeführt und mit den relevanten Technologien vertraut gemacht wurde, musste das bestehende und zu ersetzende System analysiert werden. Dies wurde in einer Ist-Analyse realisiert und die Ergebnisse (welche Funktionalität wollen wir übernehmen, welche Fehler wollen wir beheben?) dokumentiert. Basierend auf den Ergebnissen dieser Ist-Analyse wurde eine Anforderungsanalyse für das neue System durchgeführt. Die Anforderungen wurden gemäß der Spezifikation für Software-Anforderungsanalysen in funktionale und nicht-funktionale Anforderungen gegliedert. Der externen Schnittstelle zum KIM-IDM-System wurde ein eigenes Unterkapitel gewidmet. Diese Anforderungsanalyse bildet zusammen mit der darauf folgenden Modellierung der Anwendungsfalldiagramme und eines exemplarisch mittels Aktivitätsdiagramm modellierten Anwendungsfalls den Kern dieser Arbeit. Die Anwendungsfalldiagramme repräsentieren die Funktionalität des Systems bzw. der Software. Der modellierte Anwendungsfall zeigt die internen Abläufe des Systems beim Anlegen eines Accounts auf.

Um beispielhaft aufzuzeigen welche Verbesserungen das neue System mit sich bringen wird, soll hier kurz auf den Geschäftsprozess „Studentischen Account sperren/verlängern“ eingegangen werden, der in Zukunft aufgrund der vorgestellten Strukturen automatisiert werden könnte. Betrachten wir dazu zunächst das bereits in Kapitel 1.3 erwähnte studentische Attribut „Immatrikulationsstatus“. Ändert sich der Wert dieses Attributs von „immatrikuliert“ in „exmatrikuliert“ so bedeutet das, dass der Account gesperrt werden muss. Diese Änderung des Zustands des Identitätsobjekt wird dem AM-System über die Schnittstelle mit dem KIM-IDM-System zeitnah und automatisiert mitgeteilt und ermöglicht so eine schnellere Kontrolle über die Verlängerung und Sperrung studentischer Accounts. Bisher werden solche Informationen nur zum Beginn eines Wintersemesters übermittelt und müssen dann mit hohem, z.T. manuellem Aufwand abgeglichen werden.

6.2 Ausblick

In der Studienarbeit wurden alle organisatorischen und konzeptionellen Aufgaben und Aspekte erfasst und bearbeitet. Die bereits dokumentierten Anwendungsfälle müssen nun verfeinert und fertig modelliert werden. In der Studienarbeit nicht erfasste Anwendungsfälle müssen zunächst diskutiert und dann ebenfalls modelliert werden bis sie im erforderlichen Feinheitsgrad vorliegen. Um die Anwendungsfälle schon vor der Modellierung möglichst genau zu skizzieren, wird das IDM-Team diese im Gespräch mit den lokalen Administratoren der Fakultät für Informatik diskutieren. Für eine optimale Nutzung der vom KIM-IDM-System angebotenen Web Service-Schnittstelle bedarf es sowohl einer organisatorischen als auch einer technischen Abstimmung mit

dem KIM-Team. Um bezüglich des Datenschutzes alle Risiken abzudecken, wird es diesbezüglich auch Gespräche mit der Verwaltung geben. Die Ergebnisse dieser Gespräche und die vorliegende Studienarbeit bilden dann die Grundlage für die Umsetzung des neuen AM-Systems.

An diesem Punkt stellt sich die berechtigte Frage nach der Realisierung des Systems. Sollte man ein bestehendes Softwareprodukt, wie z.B. das Sun-IDM, welches auch für das KIM-IDM eingesetzt wird, nutzen? Wie hoch wäre der entstehende Anpassungsaufwand? Oder wäre es von Vorteil, das System selbst zu implementieren?

Aufgrund der individuellen Gegebenheiten und Wünsche an der Fakultät für Informatik wurde beschlossen, das System selbst zu realisieren. Die Implementierung der Software wird derzeit innerhalb der ATIS durchgeführt.

7 Anhänge

7.1 Literatur

[Ba00] Lehrbuch der Software-Technik (2.Auflage) - Helmut Balzert, Heidelberg; Berlin, 2000 (1.Auflage 1995); Spektrum Akademischer Verlag

[Be03] UML Basics: An introduction to the Unified Modelling Language – Donald Bell, IBM, Staff; 15.07.2003
www.ibm.com/developerworks/rational/library/769

[Ch06a] Understanding Web Services specifications part 1 – Nicholas Chase, 2006
http://www.ibm.com/developerworks/edu/ws-dw-ws-understand-web-services1.html

[Ch06b] Understanding Web Services specifications part 1 – Nicholas Chase, 2006
http://www.ibm.com/developerworks/edu/ws-dw-ws-understand-web-services2.html

[BR+06] Das UML Benutzerhandbuch – Grady Booch, James Rumbaugh, Ivar Jacobsen; 2006, Addison-Wesley

[HK01] Neugestaltung der Benutzerverwaltung der Fakultät für Informatik; Patrick van der Hagen und Christian Knierim, Studienarbeit an der Universität Karlsruhe, April 2001

[HS+06] Dienstorientierte Identitätsmanagement für eine Pervasive University – T. Höllrigl, A. Maurer, H. Wenske, H. Hartenstein; Karlsruhe, 2006

[HS+07] Föderatives und dienstorientiertes Identitätsmanagement im universitären Kontext – T.Höllrigl, F.Schell, H.Wenske, H.Hartenstein; Karlsruhe, 2007

[KR+03] Datenschutzaspekte von Identitätsmanagementsystemen (Recht und Praxis in Europa) – M.Hansen, H.Krasemann, M.Rost, R.Genghini; Datenschutz und Datensicherheit 27 (2003) 9

[Ku05] Identity Management – Basis für sichere Geschäftsprozesse; Martin Kuppinger, Kuppinger Cole & Partner, 2005
www.kuppingercole.de/articles/637

[Ma03] LDAP Theory and Management – Brad Marshall, Präsentation SAGE-AU Conference, 2003
quark.humbug.org.au/publications/ldap/ldap-theory.pdf

[Oe05] Die UML 2.0 – Kurzreferenz für die Praxis – Bernd Oesterreich, 2005; Oldenbourg

[OLDP] OpenLDAP Homepage, http://www.openldap.org/

[OMG] Object Management Group, http://www.omg.org/

[RFC2251] LDAPv3 – Mark Wahl, Tim Howes, Steve Kille; 1997 http://www.faqs.org/rfcs/rfc2251.html

[Ry03] Understanding Web Services – Arthur Ryman, IBM, 2003 http://www.ibm.com/developerworks/websphere/library/techarticles/0307_ryman/ryman.html

[Se05] What's new in UML 2.0? – Bran Selic, IBM, 2005 ftp://ftp.software.ibm.com/software/rational/web/whitepapers/intro_uml2.pdf

7.2 Abkürzungen

AJP	Apache JServ Protocol
AM	Account Management
ASCII	American Standard Code for Information Interchange
ATIS	Abteilung Technische Infrastruktur
DAP	Directory Access Protocol
DN	Distinguished Name
DNS	Domain Name System
HTTP	HyperText Transfer Protocol
IDBS	Institut für Dialog- und Betriebssysteme
IDM	Identity Management
IMAP	Internet Message Access Protocol
ISO	International Standards Organisation
IuK-Dienste	Informations- und Kommunikations-Dienste
J2EE	Java Platform Enterprise Edition
KIM	Karlsruher Integriertes Informationsmanagement
LDAP	Lightweight Directory Access Protocol
NIS	Network Information Service
OID	Object Identifier
OMG	Object Management Group
OSI	Open System Interconnection
POP3	Post Office Protocol 3
POSIX	Portable Operation System Interface
RDN	Relative Distinguished Name
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSH	Secure Shell
TCP/IP	Transmission Control Protocol / Internet Protocol
UB	Universitätsbibliothek
UDDI	Universal Description Discovery and Integration
UID	User Identity
UML	Unified Modeling Language
UNIX	Uniplexed Information and Computing System
URI	Uniform Ressource Identifier
VPN	Virtual Privat Network
WS	Web Service
WSDL	Web Service Definition Language
WSS	Web Service Security
XML	Extended Markup Language

7.3 Abbildungsverzeichnis

7.4 Tabellenverzeichnis