Umfassendes Autorisierungsmanagement

vorgelegt von
Diplom-Informatiker
Thomas Hildmann
aus Berlin

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr. - Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Markl
Gutachter: Prof. Dr. Kao
Gutachter: Prof. Dr. Grimm

Tag der wissenschaftlichen Aussprache: 9. Februar 2010

Berlin 2010

D 83

In Gedenken an meinen Großvater

Rudolf  Hildmann

Kurzfassung

Rollenbasierte Zugriffskontrollsysteme (RBAC) sind als Basis für ein effizientes Autorisierungssystem weit verbreitet. Diese Arbeit beschreibt einen Modellierungsansatz, der eine Identitäts-, eine Struktur-, eine Anwendungs- und eine Organisationssicht voneinander trennt, um so zu einer Verteilung von Verantwortlichkeiten zu kommen. Ferner wird ein Entwurfsverfahren für die Rollenmodellierung entwickelt, das für eine verteilte Administration geeignet ist und den Qualifizierungsaufwand sowie die Fehleranfälligkeit bei der Rollenmodellierung minimiert. Der Entwurfsprozess kombiniert Ansätze der agilen Softwareentwicklung mit Role-Engineering-Methoden und kann komplett EDV-gestützt implementiert werden.

Ein Rollenmodell dient der Strukturierung von Rollen und kann wie ein Softwaremodell behandelt werden kann. So wird beleuchtet, wie Erkenntnisse aus der Softwareentwicklung auf den Entwurf von Rollenmodellen übertragbar sind. Das vom eXtreme Programming (XP) abgeleitete Rollenentwurfsverfahren eXtreme Role-Engineering (xRE) wird ebenso wie die Anwendung von Mustern und die Modellierung über verschiedene Sichten (Views) in dieser Arbeit erörtert. Das in der Software- Entwicklung weit verbreitete Konzept der Entkopplung wird auf die Organisationsstruktur und auf die Anwendungs- und Geschäftslogik angewandt.

Die durch eine Zentralisierung entstehenden Nachteile, wie Verfolgbarkeit aller Benutzeraktionen und zentrale Datensammlungen können durch konsequente Anwendung von Methoden der mehrseitigen Sicherheit reduziert werden. Die Anwendung dieser Methodik wird am Beispiel der Authentisierung mit Smartcards, einem Modell zur nicht verfolgbaren Autorisierung sowie zur Protokollierung für verschiedene Rollen gezeigt.

Teile des in der Arbeit entwickelten Systems sind an der Technischen Universität Berlin seit einigen Jahren mit mehreren tausend Benutzern im produktiven Einsatz. Die beim Einsatz des Systems gesammelten Erfahrungen sind in die umfangreiche experimentelle Evaluation eingeflossen. Dieser Anwendungsfall zeigt die Nutzbarkeit der beschriebenen Methoden und Verfahren und bietet ferner eine Plattform für weiterführende Forschung und Entwicklung auf dem Gebiet.

Summary

Role-based access control systems as a base for efficient authorization is widely used. This work describes a model, that divides different views, such as identity, structure, user and organization to lead to a separation of duty. A role-engineering process is also developed. This is suitable for shared administration and minimizes the costs for qualification as well as it reduces error-proneness during role modeling. The engineering process combines agile software development with role-engineering methods and can be implemented completely computer aided.

A role model is a model that can be treated like a software model. This work presents ways to transfer knowledge gained from software-engineering onto role-modeling. Derived from eXtreme programming (XP) is the eXtreme role-engineering (xRE) process. xRE as well as the use of patterns and views-orientated modeling are discussed here. Uncoupling is largely used in software development and is herein applied for the organization-, the user- and business-structure.

Drawbacks arising from centralization, as traceability of user actions and central data collections, can be avoided by following the methods of multilateral security consequently. Au-

thentication via smart-cards, a model for non-traceable authorization and logging shows, the usage of this methodology exemplary for different roles.

Major parts of the system developed within this work are used daily at the Technische Universität Berlin by many thousand users since a couple of years. The gain in experience using this system led to an extensive experimental evaluation. This case demonstrates the suitability of the developed methods and processes and offers a stage for continuing research and development in this field.

Danksagung

Mein Dank gilt Prof. Odej Kao in allen seinen *Rollen* als Doktorvater. Er war für mich Motivator, Katalysator, Kritiker oder Kummerkasten ebenso wie Coautor bei Veröffentlichungen aber auch der Direktor der Zentraleinrichtung, der mir den Rahmen für diese Arbeit schaffte. Er hat in der Endphase die Betreuung meiner Arbeit übernommen und mir geholfen, "den LKW anzuschieben" (um sein eigenes Bild dafür zu verwenden) und mich dazu gebracht, die unzähligen losen Enden einzusammeln und zu einem logischen Ganzen zu verflechten.

Des Weiteren danke ich Prof. Rüdiger Grimm. Er war die Konstante in der Zeit, in der ich an meiner Dissertation gearbeitet habe. Auf ihn fiel meine erste Wahl als Zweitgutachter und er war der Erste, der zusagte, mich bei meinem Thema zu unterstützen.

Die Arbeitsgruppe von Prof. Stefan Jähnichen war es, die die Basis für den Bezug zum Software Engineering beisteuerte und er selbst war es, mit dessen Hilfe ich das Promotionsvorhaben starten konnte und der dann, als es uns beiden sinnvoll schien, die Betreuung in die Hände von Prof. Kao legte.

Das Security Team von Klaus Nagel war der Nährboden für meine Arbeiten. Er war derjenige, der jahrelang Gelder und Projekte eingeworben hat, um die Arbeit fortsetzen zu können und der mich motivierte, nicht mit der einfachsten Lösung zufrieden zu sein, sondern den eigenen wissenschaftlichen und technologieverantwortlichen Ansprüchen gerecht zu werden.

Ferner danke ich Christopher Ritter, der wohl den größten Teil des Autorisierungssystems implementiert hat. Ihm gehört der Dank dafür, dass aus den Visionen und Plänen ein reales Computersystem geworden ist.

Ich danke meinen Kollegen, die mir immer zur Seite standen, wie Gerd Schering für seine interdisziplinäre Unterstützung und Barry Linnert für die Beratung zu allen Metathemen. An dieser Stelle muss auch das gesamte ehemalige Security Team genannt werden. Ohne sie wäre diese Arbeit in dieser Form nicht möglich gewesen und hätte es den ausführlichen Erfahrungsbericht vielleicht nicht gegeben. Ich danke Euch allen für Euren Beitrag, der uns zu einem großartigen Team machte.

Mein Dank geht ferner an meine geliebte Frau, die auf viele gemeinsame Stunden verzichten musste. Die in späteren Arbeitsphasen durch diese Arbeit zur "allein erziehenden Mutter auf Zeit" wurde, die mir Mut und Trost zugesprochen hat und mir beratend und liebevoll zur Seite stand, wenn ich es brauchte.

Meinen Eltern, Schwiegereltern und Freunden danke ich jedem für seinen besonderen Beitrag, für Verständnis und Unterstützung. Ganz besonders danke ich meinem Vater für die seelische und sprachliche Hilfestellung, für einen ruhigen Platz zum Schreiben und alles, was einige schwierige Stunden etwas weniger schwierig gemacht hat.

Zuletzt geht mein Dank auch an meinen Großvater Rudolf Hildmann, dem ich diese Arbeit gerne noch persönlich vorgestellt hätte. Er lehrte mich die Freude am Schreiben, die Freiheit der Kreativität und die Pflicht des Hinterfragens und der Kritik vor dem Hintergrund einer persönlichen Vergangenheit, die in unvorstellbarem Leid resultierte und Fragen und Kritik nicht duldete und auch nur deshalb entstehen konnte, weil zu wenige die richtigen Fragen stellten und zu wenige vehement genug kritisierten.

Über diese Arbeit

Obwohl die vorliegende Arbeit als abgeschlossene wissenschaftliche Abhandlung konzipiert ist, können verschiedene Teile in sich abgeschlossen gelesen werden: Der Grundlagen Teil (Kapitel 2) kann als Überblick zu den Themenbereichen "Role-Based Access Control (RBAC)", "eXtreme Programming" und als Einblick in den Bereich der "mehrseitigen Sicherheit" genutzt werden. Das Kapitel 4, eXtreme Role-Engineering, erörtert das im Kern dieser Arbeit entwickelte Verfahren, wohingegen das Kapitel 5 Erfahrungen beim Umsetzen eines umfassenden Autorisierungsmanagements zusammenstellt und Hinweise zur technisch/architektonischen Umsetzung gibt und somit einen sehr praktischen Aspekt in die Arbeit bringt.

Diese Arbeit versucht Anglizismen zugunsten des Leseflusses und damit der Lesbarkeit so weit wie möglich zu vermeiden. Viele Fachwörter sind jedoch aus englischsprachigen Artikeln bekannt. Sofern mir eine gängige deutsche Übersetzung bekannt war, habe ich diese verwendet, einige Begriffe sind sinngemäß übersetzt, ohne dass sie mir aus deutschsprachiger Literatur bekannt waren. Um sprachlichen Missverständnissen vorzubeugen befindet sich im Anhang dieser Arbeit (Seite 145ff) ein Fachwörterbuch mit den von mir verwendeten Übersetzungen und ggf. auch synonym verwendeten Begriffen.

Neben den Fachwörtern und dem Literaturverzeichnis befinden sich im Anhang ferner ein Tabellen- und Abbildungsverzeichnis.

Bei Abbildungen, die mit "Diagrammtyp: Titel" gekennzeichnet sind, handelt es sich um UML-Diagramme gemäß der Spezifikation 2.0. Kommentare und Spezifikationen innerhalb der Grafiken sind entweder nach OCL-Notation oder nicht-formale Texte, je nachdem welche Variante leichter verständlich erschien.

Pseudocode und zu Diagrammen gehörende Texte sind jeweils grau unterlegt.

Inhaltsverzeichnis

Kapitel 1

Einleitung

> In the beginning the universe was created. This made a lot of people angry and has been widely regarded as a "bad move".
>
> D. Adams: The Hitchhikers Guide to the Galaxy - The Restaurant at the End of the Universe

Die stetig fortschreitende Digitalisierung und Vernetzung unserer Gesellschaft, vor allem jedoch unserer Arbeitswelt setzt einen zuverlässigen und effizienten Schutz der vernetzten, digitalen Güter und Dienste voraus. Die Informationstechnik befindet sich in einer Phase der Rezentralisierung von Infrastrukturen bei gleichzeitiger Modularisierung und Verteilung von Diensten. Server werden virtualisiert und auf leistungsstarke Cluster im Rechenzentrum migriert. Daten werden meist sicher im SAN gespeichert, die für die Arbeit benötigten Module liegen verteilt vor und werden in Portalanwendungen zusammengefasst. Diese Entwicklungen bedingen ein Autorisierungssystem, das in der Lage ist, eine zentrale Sicherheitspolitik auf Basis einer gemeinsamen Benutzerbasis für alle unternehmensrelevanten Dienste zur Verfügung zu stellen. Die Konfiguration der Zugriffskontrolle muss flexibel und effizient den sich ändernden Gegebenheiten angepasst werden können.

Heute eingesetzte Zugriffskontrollsysteme verwenden üblicher Weise die von Bell und LaPadula [9] geprägten Begriffe Subjekt, Objekt, Anfrage und Entscheidung. Die Entscheidung kann gemäß diesem Modell "ja", "nein" oder ein Fehler sein. Beim Subjekt handelt es sich um ein Programm, das im Sinne eines Benutzers handelt, also in der Regel dem Programm, das vom jeweiligen Benutzer ausgeführt wird. Das Objekt ist die Ressource, auf die zugegriffen werden soll und die durch das System geschützt ist. Für die Entscheidung ist ferner relevant, welche Anfrage an das Objekt gestellt wird (siehe Abb. 1.1).

Ein gängiges Zugriffskontrollmodell, das im Finanzsektor, in Unternehmen, Krankenhäusern, Universitäten usw. Verwendung findet, ist die rollenbasierte Zugriffskontrolle (RBAC). Bei der rollenbasierten Zugriffskontrolle werden Rechte nicht direkt an Benutzer vergeben, sondern an Rollen. Benutzer erhalten Rechte über Rollenmitgliedschaften (siehe Abb. 1.2).

RBAC-Modelle ermöglichen die Umsetzung verschiedener Schutzprinzipien. Dazu gehört die Verteilung von Verantwortlichkeiten und somit die Realisierung eines Mehraugenprinzips, die Definition von Nebenbedingungen, die Administration des Modells über so genannte administrative Rollen aber auch die Strukturierung des Modells über Generalisierung, Aggregation oder Supervision.

Voraussetzung für Zugriffskontrollentscheidungen ist das Identifizieren eines Subjektes. Für die Entscheidung über eine Rollenmitgliedschaft muss dem System bekannt sein, welcher Be-

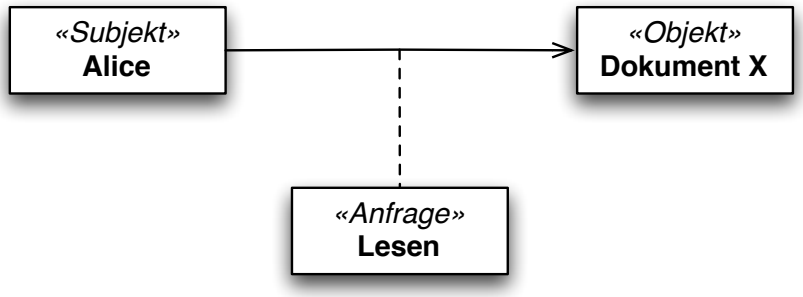

Abbildung 1.1: Objektinteraktionsgraph: Subjekt-Anfrage-Objekt

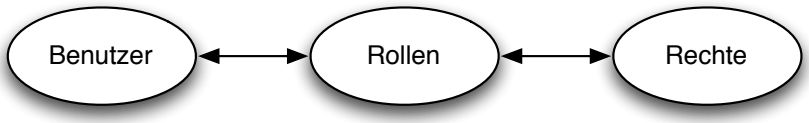

Abbildung 1.2: Beziehung zwischen Benutzer, Rollen und Rechten

nutzer den Zugriff initiiert hat. Das heißt, der Autorisierung muss eine Authentisierung vorausgehen. Für die Modellierung und Autorisierung müssen die Benutzer dem System bekannt sein. Mit dem Autorisierungsmanagement, also der Verwaltung der Zugriffe, muss demnach auch das Identitätsmanagement, also die Verwaltung potentiell berechtigter Personen einhergehen. Die Zugriffskontrolle kann je nach Sichtweise auch als Teil des Identitätsmanagements betrachtet werden.

In großen dezentral organisierten Einrichtungen oder in Föderationen aus verschiedenen Firmen oder Institutionen ergeben sich zusätzliche Anforderungen an das Autorisierungssystem. Hier muss die Autorisierung dezentral und verteilt administriert werden können. Ferner soll in solchen Szenarien oft die Sichtbarkeit in Bezug auf die einzelnen Einheiten- oder Firmeneinstellungen eingeschränkt sein. Gesetze, Verordnungen und Vereinbarungen mit unterschiedlichen Partnern und Interessenvertretern sind einzuhalten. Die Rechte, der einzelnen Parteien inklusive der Nutzer, sind zu wahren.

Es gibt verschiedene RBAC Rollenmodelle, die jeweils auf unterschiedliche Anforderungen zugeschnitten sind. Ein NIST-Standard fasst eine Familie von Rollenmodellen zusammen und bietet so eine standardisierte Basis für RBAC-Systeme. Jedoch wurden auch nach dem Erscheinen des Standards im Jahr 2000 verschiedene Erweiterungen der Modelle vorgestellt.

1.1 Das erweiterte Rollenmodell

Das in dieser Arbeit vorgestellte Modell (Abb. 1.3) ist dreigeteilt. Es besteht aus einer Identitäten-, einer Organisations- und einer Applikationsicht. Die Organisationsicht selbst besteht aus einem Organigramm oder Strukturdiagramm, also einer Abbildung der Organisation oder Föderation und den Geschäftsrollen.

Die Fragestellung, welche Rollen für den jeweiligen Anwendungsfall auf welche Weise zu

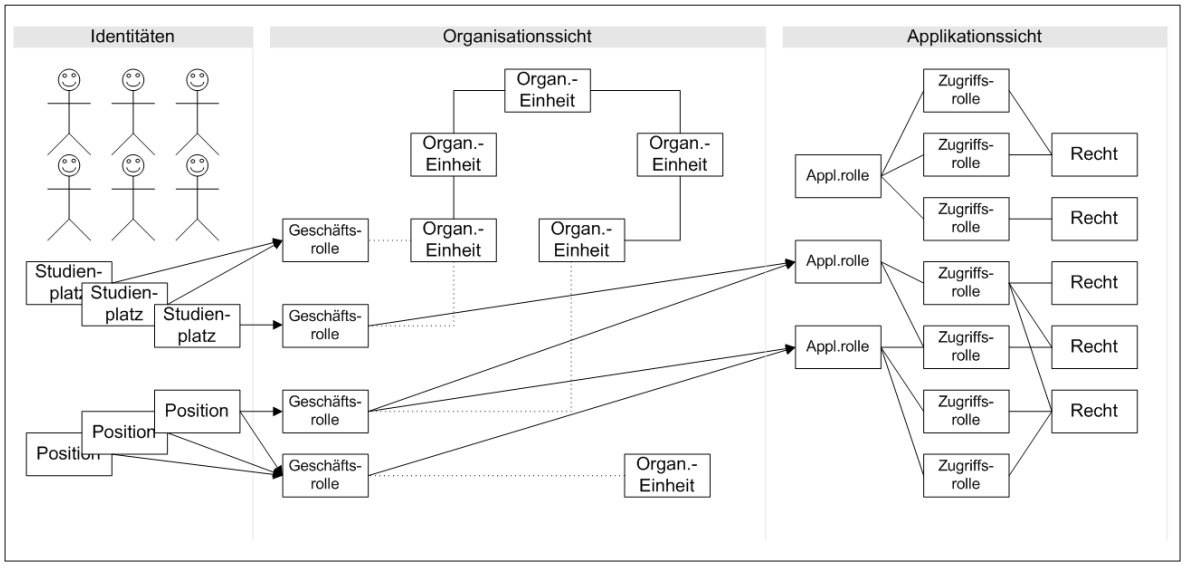

Abbildung 1.3: Vereinfachtes TUBIS-Rollenmodell

definieren sind, wird von dem so genannten Role-Engineering Verfahren behandelt. Da es sich bei dieser Aufgabe um einen kostenintensiven Prozess mit weitreichenden Folgen für die Sicherheit der IT-Systeme des Unternehmens handelt, wurde die Fragestellung der Rollenkonstruktion intensiv in der Fachwelt betrachtet und diskutiert. In dieser Arbeit wird ein neues Verfahren für das Role-Engineering vorgestellt. Klassisches Role-Engineering wurde ausgiebig in einem Buch von Edward J. Coyne [16] beschrieben.

Für den Einsatz bei der verteilten Administration von Rollenmodellen existiert ein leicht geänderter Anforderungsfokus. In einer solchen Umgebung wird ein leichtgewichtiges Role-Engineering Verfahren benötigt, das in kleinen Organisationen oder Teilen von Organisationen einsetzbar ist und das von jeweils kleinen Änderungen in der Rechteverwaltung ausgeht. Es soll den Schulungsaufwand für die Rollenadministratoren gering halten.

Die Administratoren der Einheiten benötigen ein Werkzeug, das die Zahl der definierten Rollen gering hält und bei der Definition geeigneter Rollen unterstützt. Da die Rollenadministration in solchen Szenarien in der Regel keine Haupttätigkeit darstellt, ist es wichtig, ein Verfahren zu verwenden, das den Administrator so unterstützt, dass dieser am Ende ein sicheres, korrektes und transparentes Rollenmodell erhält.

Das vor diesem Hintergrund entwickelt Verfahren nennen wir eXtreme Role-Engineering. Das Verfahren führt schrittweise durch den Rollenentwurfsprozess (Abb. 1.4). In der ersten Phase wird eine Story erstellt. Dabei handelt es sich um eine textuelle Beschreibung des neu zu implementierenden oder geänderten Anwendungsfalls (z.B. "Alle wissenschaftlichen Mitarbeiter der Einrichtung sollen den Softwarekatalog einsehen können, um Vorschläge für Anschaffung von Lizenzen machen zu können.").

Aus der Story werden Testfälle definiert. Hierzu wählt der Rollenverwalter Beispielpersonen aus (im genannten Beispiel eine Reihe von wissenschaftlichen Mitarbeitern und ggf. auch ein Negativbeispiel). Jetzt werden den Personen die Rechte zugewiesen, die sie nach erfolgter Modellierung besitzen sollen. Durch Hinzufügen, Teilen oder Kombinieren von Rollen wird nun eine neue Rollenkonfiguration erstellt. Diese Konfiguration kann schließlich in einer Sandbox (einem Simulator) getestet werden, bevor sie produktiv geschaltet wird. Stellen sich im realen

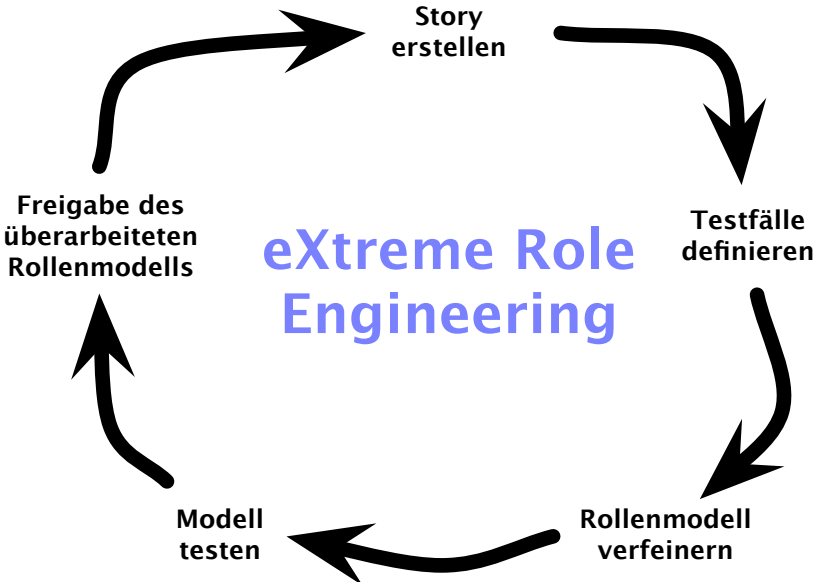

Abbildung 1.4: Der xRE-Zyklus

Einsatz nachträglich Probleme heraus, kann zum alten Zustand zurückgekehrt werden. Die Stories dienen später der Dokumentation der Änderungen am Modell.

1.2 Veröffentlichungen zur vorliegenden Arbeit

Die zu Grunde liegende wissenschaftliche Arbeit wurde in verschiedenen nationalen und internationalen Veröffentlichungen dokumentiert und auf verschiedenen Fachtagungen diskutiert: Auf den ACM Workshops on Role-Based Access Control wurden zwei Arbeiten zu Rollenmodellen und verteiltem RBAC vorgestellt [45, 30]. Ansätze zur von der Softwareentwicklung bekannten Definition von Mustern in Bezug auf RBAC-Modelle wurden in der Zeitschrift Datenschutz und Datensicherheit [31] veröffentlicht. Zur Thematik Authentisierung und Autorisierung unter Berücksichtigung von Aspekten der mehrseitigenen Sicherheit wurden auf einer Tagung der Gesellschaft für Informatik [46] (GI), auf der HPOVUA Tagung [94] und bei einer ISSE-Tagung [47] Artikel vorgestellt. Die Präsentation des eXtreme Role-Engineering Verfahrens erfolgte auf der GI-Tagung Sicherheit 2008 [48] sowie bei einem Sicherheitsworkshop des DFN [43]. Die Evaluation wurde 2007 und 2008 auf einem DFN-Forum und in der Zeitschrift Praxis der Informationsverarbeitung und Kommunikation (PIK) präsentiert [41, 49].

1.3 Gliederung der Arbeit

Die vorliegende Arbeit umfasst im groben fünf Themenbereiche. Kapitel 2 beinhaltet die Grundlagen und behandelt verwandte wissenschaftliche Arbeiten. Es werden die von dieser Arbeit berührten Themenbereiche Identitätsmanagement, Rollenbasierte Zugriffskontrolle, Role-Engineering sowie Teilbereiche des Software-Engineerings und der mehrseitigen Sicherheit betrachtet. Die wesentlichen Veröffentlichungen zu den Themenbereichen werden in Hinblick auf die Relevanz für die Arbeit zusammengefasst und in Verbindung gesetzt.

Im Kapitel 3 wird die neu entwickelte Lösung präsentiert. Dabei wird von einer Anforderungsbeschreibung auf Basis unterschiedlicher Szenarien ausgegangen. Nach einer Diskussion des Lösungsansatzes, auch im Vergleich zu existierenden Lösungen in diesem Bereich, folgt eine detaillierte Beschreibung der Lösung.

Das eXtreme Role-Engineering Verfahren wird in einem eigenen Kapitel behandelt. Da das Verfahren auch unabhängig von dem in Kapitel 3 beschriebenen System genutzt werden kann, wird in Kapitel 4 auf die speziellen Anforderungen an das Verfahren eingegangen. Es wird ein mathematisches Modell entworfen und verschiedene Berechnungsvarianten gegenüber gestellt. Darauf folgt ein Entwurf des kompletten Verfahrens. Es werden weiterführende Aspekte betrachtet, die nicht unmittelbar Bestandteil der Anforderungen sind. Das Kapitel wird mit einer Verifikation an Hand eines Prototypen und einem Vergleich des Verfahrens zu alternativen Techniken abgeschlossen.

Die Evaluation des Systems im produktiven Umfeld ist in Kapitel 5 dokumentiert. In diesem Kapitel wird auf die Implementierung des beschriebenen Systems auf unterschiedlichen Ebenen (technisch, architektonisch, organisatorisch) eingegangen. Zum Schluss werden Erkenntnisse aus dem täglichen Einsatz innerhalb der letzten Jahre erörtert.

Es folgt der Ausblick und die Zusammenfassung in den Kapiteln 6 und 7. Der Ausblick erfolgt in Form von neun kompakten Beschreibungen von möglichen Folgeprojekten. Die Schlussbetrachtung fasst die wesentlichen Erkenntnisse der Arbeit noch einmal zusammen und reflektiert über die Grenzen der Lösung.

Im Anhang befinden sich Tabellen-, Abbildungs- und Quellenverzeichnisse sowie ein Glossar der übersetzten Fachbegriffe.

1.4 Ergebnisevaluation und wissenschaftliche Einordnung

In Bezug auf das vorgestellte umfassende Autorisierungsmangement ist hervorzuheben, dass weite Teile davon an der TU Berlin seit einigen Jahren mit mehreren zehntausend Benutzern evaluiert werden. Die Evaluation geschieht im täglichen produktiven Einsatz. Zur Zeit steht nur wenig Literatur zur Verfügung, die sich mit der Umsetzung von Gesichtspunkten der mehrseitigen Sicherheit in Bezug auf RBAC beschäftigt. Die Verknüpfung von RBAC-Kriterien mit solchen der mehrseitigen Sicherheit führt zu einem leicht modifizierten Modell für die verteilte Administration des RBAC-Modells. Zur Administration dieses Modells wurde in Hinblick auf Benutzungsfreundlichkeit und Transparenz das eXtreme Role-Engineering Verfahren (xRE) entwickelt, das zur Zeit prototypisch erprobt wird und im nächsten Jahr in den produktiven Einsatz überführt werden soll.

Beim xRE handelt es sich um einen gänzlich neuen Ansatz für das Role-Engineering. Herkömmliche RE Verfahren setzen meist die Sicht auf die komplette Organisation voraus. Aus diesem Grund sind sie für viele Einsatzgebiete zu aufwändig und starr. Für einen effizienten Einsatz müssen immer Änderungswünsche gesammelt und dann in einem Modellierungsprozess verarbeitet werden. xRE hingegen begünstigt kleine, schnelle Änderungen. Möglicherweise ist xRE auch ein Impuls für die Schaffung weiterer alternativer RE-Verfahren.

Das in dieser Arbeit vorgestellte RBAC-Modell reiht sich in die Tradition der RBAC-Modellanpassungen ein. Die vorgestellten Ansätze können für verschiedene Anwendungsfälle hilfreich sein. Über den Einsatz an der TU Berlin wird nachgewiesen, dass die Verwendung in einer großen Organisation praktikabel ist.

Kapitel 2

Grundlagen und verwandte wissenschaftliche Arbeiten

> Here is a copy of everything I have. It may be of use. If anyone asks, say it fell from the sky.
>
> Delenn to Sinclair about Vorlon files in Babylon 5: "The Gathering"

Das "Umfassende Autorisierungsmanagement" baut auf der Basis bekannter Techniken auf, deren Grundlagen in diesem Kapitel dargestellt werden. In diesem Zusammenhang werden verwandte Arbeiten referenziert, die zum Vergleich oder zur Vertiefung herangezogen werden können.

Die folgenden Grundlagen sind vornehmlich für das Verständnis dieser Arbeit relevant:

- Kern ist die *rollen-basierte Zugriffskontrolle (RBAC)*. Der Schwerpunkt dieser Arbeit liegt im Bereich der Optimierung und Ergänzung von Verfahren zur rollen-basierten Zugriffskontrolle.
- Im Mittelpunkt steht das eXtreme Role-Engineering-Verfahren, das eine Kombination aus *Role-Engineering* mit adaptierten Methoden aus dem Software-Engineering im Allgemeinen und dem *eXtreme Programming* im Speziellen ist.
- Ferner wird betrachtet, inwieweit auch andere Ansätze aus dem Software-Engineering, wie z.B. *Muster*, mit in das Role-Engineering einfließen könnten.
- Zur Betrachtung sicherheitsrelevanter Themen, bei denen mehrere Parteien involviert sind, ist ferner der Themenbereich der *mehrseitigen Sicherheit* relevant.

Die rollen-basierte Zugriffskontrolle wird in vielen jüngeren, vor allem populärwissenschaftlichen Veröffentlichungen als Teil eines Identity Management Systems (IDM, auch Identity and Access Management IAM genannt) wahrgenommen. Aus diesem Grund folgt zunächst eine Vorstellung dieses Themenbereichs und eine Einordnung in dieses Umfeld.

2.1 Identitätsmanagement

"Identität" ist allgemein laut Lexikon[1] die "völlige Übereinstimmung einer Person oder Sache mit dem, was sie ist oder als was sie bezeichnet wird." Identitätsmanagement ist die Verwaltung solcher Identitäten [93]. Der umgangssprachlich unscharfe Begriff "Identität" und

[1] Das Lexikon, Zeitverlag, Hamburg 2005

die große Zahl von Anforderungen und Problemfeldern, die mit der Verwaltung von Identitäten im EDV-Umfeld einhergehen, sorgen dafür, dass sehr unterschiedliche Systeme als IDM-System bezeichnet werden [26, 8, 2, 3]. Häufig wird unter IDM jedoch die Versorgung von verschiedenen Benutzerdatenbasen für unterschiedliche EDV-Systeme durch ein zentrales System verstanden [102, 64, 49].

2.1.1 Gegenstand von IDM

Personen bewegen sich in der physischen Welt in einem soziologischen Umfeld, in dem sie in ihrer "Hauptidentität" agieren. Sie werden durch ihr körperliches Erscheinungsbild wahrgenommen und geben darüber hinaus verschiedene weiterführende Informationen (Attribute) über sich preis, wie z.B. Name, Adresse, Alter, Familienstand usw. Nicht jeder Kommunikationspartner erfährt die selben Attribute. So beschränken wir uns beim Einkauf im Supermarkt darauf, nur physisch in Erscheinung zu treten, ansonsten jedoch anonym zu bleiben; d.h. die Kassiererin würde uns vielleicht wiedererkennen, kann jedoch ansonsten nur Mutmaßungen über uns anhand unserer Einkäufe anstellen. Zahlt eine Person jedoch mit seiner EC-Karte, so werden Name und Kontoverbindung bekanntgegeben. Da eine Person jeweils nur Teile ihrer Identität preisgibt, spricht man von Teilidentitäten, mit denen die Person arbeitet.

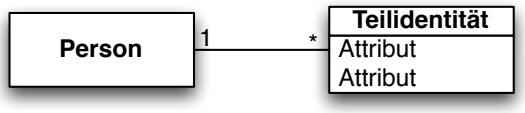

Abbildung 2.1: Klassendiagramm: Gegenstand des Identitätsmanagements

In der digitalen und in der physischen Welt benötigt man für unterschiedliche Anwendungen verschiedene Teilidentitäten, d.h. unterschiedliche Attribute der Person. Gegenstand des IDM ist die Zuordnung von n Teilidentitäten zu 1 Personen (siehe Abb. 2.1).

2.1.2 Ziele des IDM

Ziel eines Identitätsmanagements sind konsistente, verfügbare und verlässliche (im Sinne von Echtheit) Daten zu Personen in allen in das IDM integrierten EDV-Systemen.

Konsistenz: Die Attribute zu allen Teilidentitäten einer Person müssen zu jedem Zeitpunkt widerspruchsfrei sein. Viele Attribute ändern sich im Laufe der Zeit, wie z.B. Namen bei Heirat, Adressen bei einem Umzug, Bürotelefonnummern usw. Das IDM muss Möglichkeiten zur Änderung dieser Attribute zur Verfügung stellen und die geänderten Daten an die jeweiligen Nutzer der Daten (z.B. Webportal, Rechnerzugang, E-Mailaccount) verteilen (siehe Testszenarien in [90]).

Verfügbarkeit: Die Attribute zu den vom IDM verwalteten Personen müssen den jeweiligen Nutzern der Daten jeder Zeit zur Verfügung stehen. Dies stellt hohe technische, wie organisatorische Anforderungen an ein IDM.

Echtheit: Die Daten müssen eine der jeweiligen Anwendung angemessene Qualität besitzen. Sowohl bei der Erfassung wie auch bei den Änderungen muss sichergestellt sein, dass

die jeweilig erfassten Attribute echt sind. Was Echtheit im jeweiligen Anwendungsfeld bedeutet und wie hoch die Anforderungen an die Qualität der Daten sind, hängt vom bestimmten Anwendungsszenario ab (vergl. z.B. Webföderation einer Hobbygemeinschaft und Intranet einer Anwaltskanzlei).

Folgende Problemfelder spielen dabei eine besondere Rolle:

Geltungsbereich: Welche Attribute sollen in welchem Umfeld verfügbar sein; d.h. welche Anwendungen sollen mit welchen Daten versorgt werden? Umfasst eine Föderation evtl. auch organisationsübergreifende Dienste? Wie sind die Anforderungen an die Datenqualität innerhalb dieser Föderation?

Lebenszyklus: Durch die Phasen des gesamten Lebenszyklus der Daten, müssen diese konsistent gehalten werden. Dazu zählen: Einrichtung, Modifikation, Suspendierung und Terminierung bzw. Archivierung der Daten.

Verwaltung und Schutz der Informationen: Es liegt in der Natur der Verwaltung von Identitäten, dass es sich in der Regel um sensible und/oder zu schützende Daten handelt. Neben den zu respektierenden Anforderungen durch die im IDM geführten Personen, sind meist auch rechtliche Rahmenbedingungen einzuhalten, wie die jeweils geltenden Datenschutzbestimmungen. Dieses Problemfeld wird im Zusammenhang mit IDM "Compliance" genannt [2, Stichwort: Compliance, 11.10.2006].

Rollen und Identitäten: Rollen und Rechte müssen den Identitäten zugeordnet werden, d.h., dass Mitgliedschaften in Rollen definiert werden müssen. Hierbei ist dafür Sorge zu tragen, dass die Zuordnung gemäß der vorab definierten Sicherheitspolitik stattfindet, da eine Zuordnung in falsche Rollen sowohl zu viele, als auch zu wenige Rechte zur Folge haben kann [64].

Datenspeicherung: Es sind Entscheidungen darüber zu treffen, auf welche Attribute über welche Datenquellen zugegriffen werden kann. Dabei werden Informationen oft verteilt. Ein Beispiel hierfür ist der Einsatzes von Tokens, wie z.B. Smartcards, die dann den Nachweis über die Identität (Authentisierung) einer Person liefern können. Der Benutzer wird hierbei in die Lage versetzt, einen Teil der Daten mit sich zu führen.

2.1.3 Architekturelle Bestandteile

Die Softwarearchitektur hinter IDM-Systemen ist divergent. Die meisten Architekturen lassen jedoch folgende Bestandteile erkennen (Abb. 2.2):

Provisionierungsmodul: Bei der Provisionierung (engl. Provisioning) wird der Initialzustand der Teilidentitäten zu einer Person im System erzeugt. Dazu wird ein Satz an Stammdaten erfasst und im IDM-System verteilt. Die Datenerfassung muss hierbei jeweils wieder die Anforderungen an die Datenqualität erfüllen.

Access-Management: Die Bereitstellung von geeigneten Zugriffsrechten auf die jeweiligen EDV-Systeme wird vom Access-Management gesteuert. Auf diesem Bestandteil des IDM liegt der Fokus meiner Arbeit.

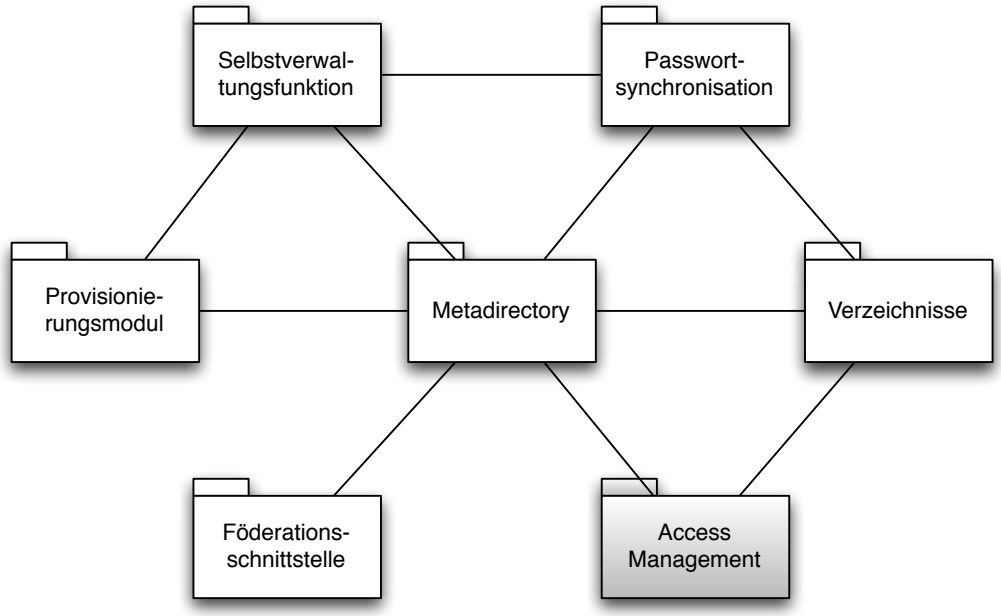

Abbildung 2.2: Komponentendiagramm: Klassische Komponenten eines IDM Systems

Metadirectory: Beim Metadirectory handelt es sich aus Software Engineering-Sicht um eine Broker-Architektur, die die Attributanforderungen der Zielsysteme auswertet und die jeweils tatsächlichen Datenquellen kapselt. Die Informationen zu einer Person können also in verschiedenen Primärquellen vorliegen und werden vom Metadirectory für die Anwendungen zusammengestellt.

Verzeichnisse: Der LDAP-Verzeichnisdienst spielt im Bereich IDM eine zentrale Rolle. Viele Anwendungen können ein LDAP-Verzeichnis als Datenbasis nutzen. Das Active-Directory von Microsoft ist eine Variante hiervon und spielt auf Grund der weiten Verbreitung von Microsoft-Produkten eine wichtige Rolle. Es existiert jedoch auch eine Vielzahl weiterer Datenbanken für Benutzerdaten, die jeweils vom IDM konsistent mit Daten versorgt werden müssen.

Föderationsschnittstelle: Soll das IDM Teil einer Föderation werden, so müssen geeignete Schnittstellen bereitgestellt werden. Eine besondere Rolle spielt in diesem Zusammenhang vor allem im universitären Umfeld das Shibboleth-System. Zur Zeit kann jedoch noch nicht behauptet werden, es hätte sich ein Standard für IDM oder auch nur bei der Föderation durchgesetzt.

Passwortsynchronisierung: Die Verwendung eines einzigen organisationsweiten Passworts stellt einen Sonderfall der Datenverteilung dar. In der Regel stellen IDM-Systeme hierfür Webschnittstellen zur Verfügung. Passwörter werden in der Regel nicht über das Metadirectory abgefragt, sondern vom Passwortsynchronisationsmodul auf den Zielsystemen gesetzt. Dabei müssen bei der Passwortfestsetzung Regeln gefunden werden, die die Anforderungen aller Zielsysteme gleichermaßen berücksichtigen.

Selbstverwaltungsfunktionen: Neben einer Benutzungsschnittstelle zur Passwortänderung können auch weitere Funktionen für die Benutzer zur Einsicht und Änderung ihrer Teilidentitäten und damit verknüpften Attributen angeboten werden.

2.1.4 Benutzerorientierte Sicht auf IDM

Unter Identitätsmanagement wird auch die technische Verwirklichung des Rechts auf informationelle Selbstbestimmung durch Nutzer von Online-Diensten verstanden [58].

Oder, wie in der Studie [26] ausgedrückt: *There are some user-oriented definitions of identity management. A very impressive slogan characterises privacy-enhancing identity management: "Make the user owner of her profile."*

Hinter dieser Familie von IDM-Systemen stehen zwei grundsätzliche Probleme bei dem Umgang mit Identitäten im Internet:

1. Die Verfolgbarkeit von Personen anhand der von ihnen hinterlassenen Datenspuren durch Angabe von personenbezogenen Daten, IP-Nummern, Cookies etc.

2. Die mangelnde Überprüfbarkeit von Personendaten bei Online-Transaktionen.

Der folgsame Konsument verliert seine Privatsphäre, wohingegen Identitätsdiebstahl, d.h. Missbrauch von fremden Identitäten, nicht verhindert werden kann oder sogar durch immer vollständigere Datenspuren gefördert wird.

Identitätsmanagement in diesem Sinne stellt Möglichkeiten zur Verfügung, Benutzer anonym oder pseudonym zu identifizieren und dem Benutzer transparent zu machen, bei welchem Dienstanbieter welche identitätsbezogenen Daten vorliegen, welche Kommunikationspartner also über welches Wissen verfügen.

Auch in diesem Zusammenhang wird wieder von Rollen gesprochen. Diese sind als Funktionsrollen im sozialen Umfeld zu verstehen. Eine Person kann "Bankkunde", "Vater", "Kollege", "Leser", "Mitspieler" usw. sein. In Abhängigkeit von seiner Rolle gibt das IDM-System die vorher in Profilen zusammengestellten Daten preis. Die Einordnung in Rollen hilft dabei dem Benutzer zu entscheiden, welche Informationen wirklich nötig sind. Um ein Weblog zu lesen ("Leser") ist mein Name ebenso unwichtig, wie meine E-Mailadresse. Beim Hinterlassen von Kommentaren in einem Forum kann ein Pseudonym helfen, die Mitteilungen einer Person später beim Lesen im Zusammenhang zu betrachten. Hier ist der Benutzer gefragt zu entscheiden, ob er Pseudonymität bzw. Anonymität bevorzugt oder welche seiner Teilidentitäten er zu nutzen wünscht.

In der vorliegenden Arbeit verstehe ich "datenschutzgerechtes IDM" im Sinne eines wie in Abschnitt 2.1 beschriebenen Systems unter Berücksichtigung datenschutzrechlicher Aspekte und unter Berücksichtigung der Ziele der mehrseitigen Sicherheit 2.6.

2.2 Rollenbasierte Zugriffskontrolle (RBAC)

Zugriffskontrolle existiert in vielen unterschiedlichen Bereichen überall dort, wo es darum geht, ein spezielles Gut zu sichern. Im Folgenden wird die Zugriffskontrolle in IT-Systemen und das rollenbasierte Zugriffsmodell im Speziellen betrachtet.

2.2.1 Zugriffskontrolle in IT-Systemen

B. W. Lampson führte die Begriffe "Subjekt" und "Objekt" in Bezug auf Zugriffe in Computersysteme ein [61]. Dabei bezeichnet man jedes Computerprogramm, das auf Anweisung einer Person handelt, als "Subjekt". Alle direkt oder indirekt von einem Benutzer gestarteten Programme fallen in diese Kategorie. "Objekte" sind die Ressourcen auf einem Computersystem, wie Peripheriegeräte, Netzwerkverbindungen oder Datensätze. Dabei ist die Granularität der Betrachtung, allein von den Anforderungen der jeweiligen Anwendung abhängig. Also die Frage danach, ob ganze Dateisysteme, Verzeichnisinhalte, Dateien, einzelne Datensätze oder ein einzelnes Datum als Objekt im Sinne der Zugriffskontrolle verstanden wird. Diese Betrachtung ermöglicht die Erstellung einer Zugriffsmatrix, die widerspiegelt, welche Operationen von gegebenen Subjekten auf bestimmte Objekte gestattet sind.

Tabelle 2.1: Beispiel für eine Zugriffsmatrix

	Webportal	Bestellungen	Prüfungsnoten
Alice	lesen	-	lesen
Bob	lesen, schreiben	lesen, schreiben	-
Charles	lesen	-	lesen, schreiben

Die Tabelle 2.1 zeigt ein Beispiel für eine Zugriffsmatrix. "Alice", "Bob" und "Charles" repräsentieren die Subjekte, "Webportal", "Bestellungen" und "Prüfungsnoten" die Objekte dieses Beispiels. Aus der Tabelle kann jeweils abgelesen werden, welche Art von Zugriff die jeweiligen Subjekte auf die Objekte haben. So darf ein Programm, das im Auftrag von Alice arbeitet, beispielsweise nur Prüfungsnoten lesen, wohingegen ein Programm von Charles auch zum Schreiben von Prüfungsnoten berechtigt ist.

Beim Bell-LaPadula-Modell [9] handelt es sich um ein mathematisches Modell für mehrstufige Sicherheit, wie sie hauptsächlich beim Militär Anwendung findet. Dabei werden Subjekte und Objekte klassifiziert. Jeder Benutzer und jedes Objekt bekommt eine Freigabestufe ("vertraulich", "geheim", "streng geheim"). Die Grundregel besagt nun, dass jedes Subjekt Zugriff auf die Objekte mit gleicher oder niedrigerer Klassifizierung hat. Neben dieser einfachen Grundregel kennt das Modell ferner eine vertikale Einteilung in Kategorien (z.B. "Atomwaffen", "NATO") und eine so genannte "*-Eigenschaft", die das Schreiben in höhere Klassifizierungen ermöglicht. Diese Eigenschaft wird für die Implementierung von Verwaltungsprogrammen in Computersystemen nötig, da sich diese keine Informationen "merken" und nach dem Anmelden in einer höheren Ebene wieder eingeben können.

Die beiden heute noch extrem weit verbreiteten Zugriffsmodelle für Computer sind Discretionary Access Control (DAC) und Mandatory Access Control (MAC). DAC basiert auf dem Grundprinzip, dass der Zugriff auf Objekte von Eigentümern der Objekte verwaltet wird. So besitzt ein Benutzer eine Menge von Dateien, auf die er Zugriffe durch andere Benutzer steuern kann. Das erreichte Sicherheitsniveau ist so durch die Benutzer und durch ihr Verhalten bezüglich Zugriffsverwaltung der ihnen anvertrauten Objekte bestimmt. Im Gegensatz hierzu ist die Zugriffsverwaltung im MAC den Benutzern vorgegeben. Üblich ist eine Implementierung mittels mehrstufiger Sicherheit, wie von Bell-LaPadula dargelegt.

Schwierigkeiten in der Implementierung eines geeigneten Sicherheitsmodells im Umfeld der kommerziellen Nutzer brachte Überlegungen hervor, deren Schwerpunkt weniger auf der Ver-

traulichkeit, als viel mehr auf der Integrität der Daten lag. Das Modell von Clark und Wilson [14] legt daher Wert auf gültige Transaktionen (well-formed Transactions) und die Verteilung von Verantwortlichkeiten.

2.2.2 Authentisierungsmethoden

Die Zugangskontrolle bei IT Systemen verhindert die Nutzung durch unautorisierte Personen. Üblich ist eine Authentisierung, bei der der Benutzer zunächst seine Identität angibt, die er dann durch Wissen, Besitz oder biometrische Merkmale beweist [35] (siehe Abb. 2.3).

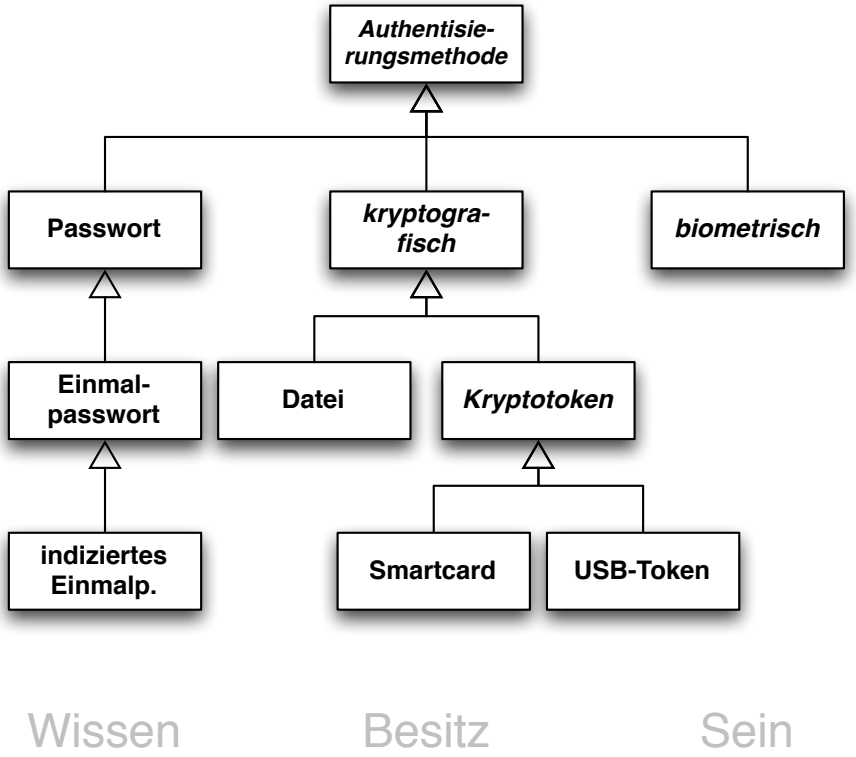

Abbildung 2.3: Klassendiagramm: Authentisierungsmethoden

Benutzername/Passwort: Der Benutzer gibt auf einer Login-Webseite seinen Benutzerkontennamen und ein dazu korrespondierendes Passwort ein. Dies ist die älteste und schwächste Authentisierungsmethode. Sie ist anfällig gegen organisatorische Angriffe, wie die Weitergabe von Passwörtern oder ein offensichtliches Hinterlegen aber auch gegen entfernte Angriffe. Selbst wenn Wörterbuchattacken durch Regeln zur Passwortwahl unterbunden werden, so sind es genau diese Regeln, die den Schlüsselraum für Passwörter massiv einschränken. Gepaart mit schnellen Netzwerkverbindungen und verteilten Angriffen, wie sie heute an der Tagesordnung sind, stellen Passwörter heute keinen ernstzunehmenden Schutz mehr dar. Die Tatsache, dass sie noch immer eingesetzt werden, ist der Einfachheit des Mechanismus geschuldet und dem Umstand, dass Passwörter überall genutzt werden können, wo es die Möglichkeit zur Buchstaben- und Zahlenübermittlung gibt, also auch am Internetterminal in der Lobby, dem Smartphone oder dem PC eines Kollegen. Aber gerade diese Möglichkeiten erhöhen die Gefahr

des Ausspähens durch Keylogger, Historyfunktionen oder einfach nur dem Schulterblick einer im Raum befindlichen Person.

Einmalpasswörter: Zusätzlich zur sehr schwachen Authentisierung durch Benutzername und Passwort können Einmalpasswörter genutzt werden, um die meisten Schwachpunkte einer bloßen Passwortauthentisierung auszugleichen. Ein Ausspähen ist hier nicht zielführend, vor allem, wenn es sich um ein indiziertes Verfahren, wie beispielsweise der iTAN handelt. Die Weitergabe der TAN-Liste wird selbstverständlich technisch nicht verhindert und eine entwendete, abfotografierte oder auf anderem Wege kopierte TAN-Liste kann leicht von Angreifern genutzt werden. Die Nutzung von TAN-Listen wird von Benutzern als lästig empfunden und kostet Zeit im Arbeitsablauf.

Smartcards/Tokens: Da eine Authentisierung gegen Schlüsselpaare auf Massenspeichern ebenfalls die Gefahr der gewollten oder ungewollten Weitergabe birgt, ist die PKI-basierte Authentisierung mittels Smartcards oder Tokens (Smartcards auf USB-Sticks) das State-of-the-Art Authentisierungsverfahren [79]. Der Verlust einer Smartcard kann durch automatische Sperrung bei mehrmaliger PIN-Falscheingabe gesichert werden und selbst Keylogger können einzig in Besitz der PIN kommen, die ohne zugehörige Smartcard nutzlos ist. Zwar kann selbst dies durch Verwendung von Smartcardlesern mit Tastenfeld unterbunden werden. Nur scheitert dieses Verfahren an massiver Unterspezifizierung der entsprechenden Standards und an mangelnder Nachfrage durch die Benutzer, so dass kaum eine funktionierende Smartcardlösung mit PIN-Padunterstützung erhältlich ist.

Biometrische Merkmale: Biometrische Merkmale erfreuen sich zur Zeit wachsender Beliebtheit, da zum Nachweis der Identität hierfür allein die messbaren Eigenschaften einer Person notwendig sind. Neben den bekannten Fingerabdruck- und Retinascans, sind auch Handschriftproben, Stimmproben oder sogar Bewegungsmuster beim Laufen verwendbar. Auch Handvenenmuster oder Hand- bzw. Gesichtsgeometrien können zum Vergleich herangezogen werden. Viele Biometrische Verfahren haben heute noch Schwächen bei natürlichen Schwankungen der Merkmale, wie z.B. Heiserkeit bei Stimmproben, leichten Hautverletzungen beim Fingerabdruckscan usw. Eine weitere Hürde bei der Einführung von biometrischen Authentisierungsmethoden ist die Schaffung von Barrieren für Personen mit vom Standard abweichender Physiologie (Netzhauttrübung, Stummheit, usw.).

2.2.3 X.821 Access Decision Facility und Access Enforcement Facility

Architektonisch werden zwei für die Zugriffskontrolle elementare Komponenten unterschieden: Die Zugriffsentscheidungseinheit (ADF - Access Decision Facility) und die Entscheidungsdurchsetzungsinstanz (AEF - Access Enforcement Facility[2]). Die AEF-ADF-Architektur ist in der X.821 Empfehlung [1] zu finden. Ein Ziel wird durch eine AEF geschützt. Der Zugriffswunsch eines Initiators wird von der ADF geprüft. Für die Entscheidung kann die ADF neben den Basisinformationen (Subjekt, Anfrage, Objekt) auf Kontextinformationen, wie z.B. die IP-Adresse des Initiators und auf externe Informationen, wie z.B. die Zeit zurückgreifen. Die Entscheidung wird gemäß einer Zugriffskontrollinformationsdatenbank (ACI - Access Control Information) getroffen.

[2]Englische Fachbegriffe werden in der Regel übersetzt. Im Anhang befindet sich eine Gegenüberstellung der verwendeten Übersetzungen und der in der englischsprachigen Literatur üblichen Vokabeln.

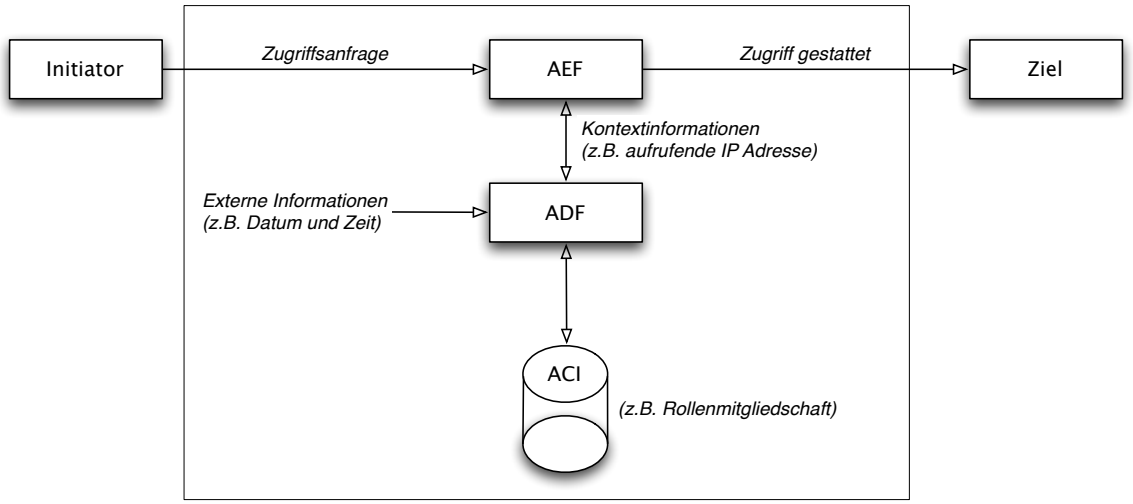

Abbildung 2.4: AEF/ADF-Architektur nach X.821

Es gibt verschiedene Zugriffskontrollmodelle, die in einer ACI gespeichert und von der ADF ausgewertet werden können.

2.2.4 RBAC-Modell

In der Einleitung zu [24] heißt es sinngemäß: *"Aus einer Unternehmenssicht hat Zugriffskontrolle das Potential, den optimalen Austausch und das Verteilen von Ressourcen zu fördern; es hat jedoch auch das Potential, Benutzer zu frustrieren, große administrative Kosten zu erzeugen und unautorisierte Enthüllungen oder Fälschungen von sensiblen Informationen zu begünstigen."*

Rollenbasierte Zugriffskontrolle entstammt ursprünglich ebenfalls dem gewerblichen Umfeld. Das Grundprinzip der rollenbasierten Zugriffskontrolle [23] besteht darin, dass Benutzer jeweils Mitglieder in Rollen sind. Rechte und Pflichten sind dann jeweils an Rollen und nicht direkt an Personen gebunden (siehe Abbildung 1.2). Das Prinzip der Verantwortlichkeiten für unterschiedliche Bereiche und damit das Agieren in einer bestimmten Rolle ist im wirtschaftlichen Kontext weit verbreitet. Das Erstellen eines Sicherheitsmodells gemäß RBAC stellt daher eine Vereinfachung für die Bereiche dar, die sich nicht leicht auf das eher militärische Mehrstufenmodell oder die Verwaltung durch die Benutzer selbst abbilden lassen oder bei denen es nicht akzeptabel ist, den Sicherheitszustand von der Rechtevergabe der Benutzer allein abhängig zu machen.

Tabelle 2.2: Rollenmodelltypen nach [85]

$RBAC_0$	Basis RBAC-Modell, das dem Prinzip der geringsten Rechte folgt
$RBAC_1$	Fügt dem $RBAC_0$-Modell Rollenhierarchien hinzu.
$RBAC_2$	Erweitert $RBAC_0$ um Nebenbedingungen, wie z.B. sich ausschließende Rollen
$RBAC_3$	Enthält die Eigenschaften von $RBAC_1$ und $RBAC_2$

Basis der heute gängigen allgemeinen Rollenmodelle sind die von Sandhu [85] vorgestellten Modelltypen (siehe Tabelle 2.2), auf die sich auch verschiedene Erweiterungen zurückführen lassen [84, 77, 63, 39]. Auch der von Jaeger [50] kritisierte NIST-Standard [22] zu RBAC basiert auf den 1996 von Sandhu vorgestellten Modelltypen.

RBAC-Systeme sind in sehr unterschiedlichen Produkten auf verschiedenen Anwendungsschichten implementiert. Es gibt RBAC-Systeme, die auf Betriebssystemebene greifen, Rollenmodelle in Datenbankmanagementsystemen, in Anwendungen bis hin zu unternehmensweiten Modellen, bei denen die Konfigurationen für unterschiedliche Zugriffskontrollen auf die Zielsysteme verteilt werden.

2.2.5 Dezentrale Administration

Das Prinzip der Delegation von Aufgaben wird in großen Organisationen angewandt, um der Tatsache Rechnung zu tragen, dass es nicht möglich ist, alle Aufgaben von einer Person bewältigen oder alle Personen alle Aufgaben in gleichem Maße durchführen zu lassen. Der Einsatz einer verteilten Rechteverwaltung für die IT-Infrastruktur ist hierbei ein logischer Schritt, der eine flexible, schnelle und transparente Anpassung der Prozesse innerhalb einer Organisation von der Basis her ermöglicht (siehe Abb. 2.5 und Tabelle 2.3).

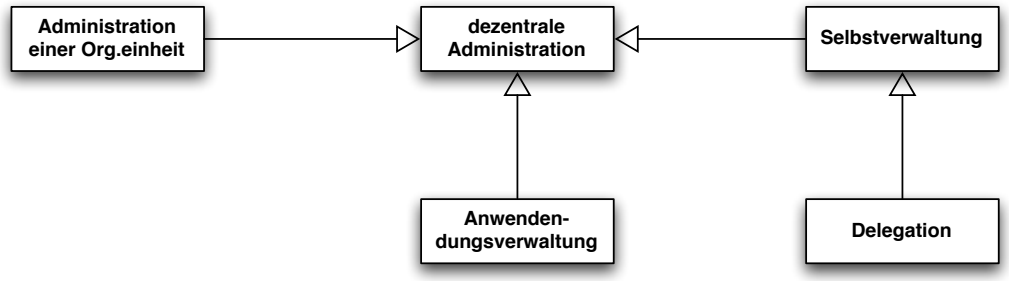

Abbildung 2.5: Klassendiagramm: Klassen dezentraler Administration

Tabelle 2.3: Klassen dezentraler Administration und Wirkungsbereich

Administrationsklassen	Wirkungsbereich
Administration einer Organisationseinheit	BENUTZER-ROLLE, ROLLE-ROLLE
Anwendungsverwaltung	ROLLE-RECHT
Delegation	BENUTZER-ROLLE
Selbstadministration	BENUTZER-*

2.2.6 Schutzmechanismen in RBAC-Modellen

Mit der Möglichkeit der verteilten Administration ist die Gefahr verbunden, die Sicherheitspolitik der Organisation vorsätzlich oder versehentlich zu verletzen. Für die Minimierung von

Risiken in RBAC-geschützten Umgebungen stehen verschiedene Maßnahmen zur Verfügung, die im Folgenden mit ihrer jeweiligen Wirkungsweise aufgezeigt werden.

Viele dieser Maßnahmen sind für RBAC-Systeme optional. In [85] wird ein Basismodell $RBAC_0$ definiert, in dem es weder Nebenbedingungen (Constraints) noch Hierarchien gibt. Erst das $RBAC_3$ Modell vereinigt alle diese Eigenschaften. Vor dem Hintergrund der Sicherheit wird hierdurch klar, welchen Stellenwert die Wahl eines geeigneten Zugriffsystems mit adäquater RBAC-Implementierungen hat.

Verteilung von Verantwortlichkeiten

Ein sehr mächtiges Konzept zum Schutz der Ressourcen aber auch zur Entlastung der Benutzer ist die Verteilung der Verantwortlichkeiten (SoD, Separation-of-Duty) [23]. Die Konzepte hierzu sind aus der physischen Welt in die IT-Modelle übernommen worden. So sind papierbasierte Arbeitsabläufe hinreichend bekannt, bei denen eine zweite oder dritte Person nach Prüfung des Sachverhalts ebenfalls unterzeichnen muss (Vier-Augen-Prinzip) oder wo die Verteilung von Ämtern sich gegenseitig ausschließt. Das in der RBAC-Literatur immer wieder zitierte Beispiel ist der Kassenwart, der nicht seine eigene Kasse prüfen darf.

Bedürfen kritische Operationen die Zusammenarbeit mehrerer Benutzer [14], können so Kompromittierungen des Systems durch falsche Rollenzuordnungen in den meisten Fällen verhindert werden.

Die Verteilung der Verantwortlichkeiten kann in RBAC-Systemen auf sehr unterschiedlichen Ebenen umgesetzt werden. Man unterscheidet hier u.a. statische (SSD) und dynamische (DSD) SoD.

Das SSD greift bei der Zuordnung von Personen zu Rollen. Ist eine Person Mitglied in einer bestimmten Rolle, die die Mitgliedschaft in einer anderen Rolle ausschließt, dann kann die Person der zweiten Rolle nicht zugewiesen werden.

Im Gegensatz hierzu greift die DSD erst bei der Aktivierung der Rolle, also dann, wenn sich der Benutzer mit der Rolle anmeldet. An dieser Stelle muss sich der Benutzer entscheiden, welche der Rollen er benutzen will oder umgangssprachlich: "Welchen Hut er bei der Erledigung seiner Aufgaben auf hat." Es ist zu jeder Zeit möglich, sich mit einer der Rollen ab- und mit einer anderen anzumelden.

Die operationale SoD wiederum basiert auf der Tatsache, dass zur Durchführung einer bestimmten kritischen Funktion die Rechte mehrerer Rollen notwendig sind. Wird ferner über die o.g. Mechanismen dafür gesorgt, dass eine Person nicht gleichzeitig im Besitz aller nötigen Rollen ist, kann damit sichergestellt werden, dass die Ausführung dieser kritischen Funktion nur durch mehrere Personen autorisiert werden kann.

In der Literatur ist ferner von "history and object-based SoD" die Rede. Diese Art der Verantwortlichkeitsverteilung ist jedoch nicht weit verbreitet. Sie basiert darauf, dass eine Person zwar alle Rollen und damit Rechte gleichzeitig besitzen kann, sie jedoch nicht auf ein und dasselbe Objekt anwenden darf. So darf ein Kassenwart auch Kassenprüfer sein. Wenn er jedoch eine bestimmte Kasse geführt hat, so darf er diese nicht prüfen.

Grundsätzlich ist SoD sowohl für die Umsetzung einer Sicherheitspolitik ein sehr mächtiges Werkzeug, als auch für die Benutzer. Verhindert man die "Allmacht" von Administratoren, so können diese nicht in den Verdacht geraten, unautorisiert tätig geworden zu sein. Verteilung von Verantwortung geht auch immer mit Verteilung von Aufgaben und damit von Arbeit einher. SoD kann also dazu dienen, die Arbeitsverteilung im Sicherheitsmodell der Organisation abzubilden.

Administrative Rollen

Die Administration des Rollensystems kann selbst wieder über "administrative Rollen" gesteuert werden [84]. Diese Rollen besitzen Rechte, die die Veränderung des Rollenmodells zulassen. Insbesondere wird die Zuordnung von Benutzern zu Rollen häufig verteilt, weil dies in RBAC-Systemen den größten Aufwand im Betrieb darstellt. Im IAM Wiki Lexikon [2] wird zwischen "Rollenadministrator" und "Rollenzuteiler" unterschieden. Wobei der Rollenzuteiler die beschriebene Zuteilung von Benutzern zu Rollen durchführt, wohingegen der Rollenadministrator Rollen definiert und diese mit Rechten ausstattet.

Die Verteilung der Rollenadministration hat ferner den Vorteil, dass Blockierungen verhindert werden können, die dazu führen, dass Personen auf Grund fehlender Rechte nicht arbeitsfähig sind.

Es gibt unterschiedliche Modelle, die den Umfang der administrativen Rechte definieren, d.h. die Frage, welche Teile des Modells von welchem Administrator verwaltet werden dürfen.

Die Vergabe von administrativen Rollen hat verschiedene Vorteile: Die Rolle kann an Nebenbedingungen geknüpft werden. So gibt es z.B. an der Universität die Rolle "Dekan". Der Dekan wird von einem Gremium gewählt. Nach Wahl eines neuen Dekans, verliert der amtierende Dekan seine Rolle und der neue Dekan wird Mitglied. Damit gehen alle administrativen Rollen, die an der "Dekan"-Rolle hängen, automatisch mit verloren. Die Reduzierung des Verwaltungsaufwands an dieser Stelle erhöht weiterhin die Sicherheit. Vom Zeitpunkt des Entzugs der Rolle steht der ausgeschiedene Dekan ferner nicht mehr in der Verantwortung für seinen Bereich.

Allgemeine Festlegungen durch Nebenbedingungen

Nebenbedingungen (Constraints) können in RBAC-Modellen dazu verwendet werden, um im Modell einige Grundregeln zu verankern. Nebenbedingungen können theoretisch an allen Stellen im Modell definiert werden, sofern das von der Implementierung unterstützt wird:

- Bei der Person-Rollen-Zuordnung,
- der Rollen-Rollen-Zuordnung (siehe Abschnitt 2.2.6),
- bei der Rollen-Rechte-Zuordnung,
- aber auch bei der Aktivierung von Rollen,
- oder der Nutzung von Rechten.

Die Nebenbedingungen können dazu genutzt werden, um Sachverhalte auszudrücken, die mit der bloßen Zugehörigkeit zu einer Rolle nicht definierbar sind. Sie können aber auch dafür verwendet werden, um sicherzustellen, dass die Organisation handlungsfähig bleibt. So können Regeln definiert werden, die z.B. verhindern, dass bestimmte Rollen unbesetzt sind, dass sie nur ein oder x-mal besetzt werden können oder die besagen, was im Falle einer Nichtbesetzung geschieht.

In einigen Organisationen gibt es beispielsweise eine Rolle, die besagt: Ist die Rolle des Rollenverwalters nicht besetzt, so kann der Rollenverwalter der strukturell übergeordneten Einheit die Untereinheit verwalten, bis dort ein Verwalter eingesetzt ist. Hier greift eine Nebenbedingung auf eine Rollen-Rollen-Zuordnung. Der Rolle Rollenverwalter sind wiederum

andere Rollen zugeordnet, die das Verwalten ermöglichen. Grundsätzlich besitzt jeder Rollenverwalter auch die Rollen aller Untereinheiten. Diese Zuordnung geht jedoch verloren, sobald auf einer Unterebene ein anderer die Verwalterrolle besitzt. Dem Verwalter der Untereinheit ist es nun freigestellt, die Verwaltungsrolle wieder explizit an den übergeordneten Verwalter zu übertragen. Faktisch ist dies jedoch selten der Fall.

Über eine einfache Nebenbedingung kann so also beispielsweise verhindert werden, dass eine Untereinheit handlungsunfähig wird. Auf die gleiche Weise können über Nebenbedingungen auch andere Grundregeln der Sicherheitspolitik verankert werden. Diese können dann auch nicht durch fehlerhafte Zuordnungen im Modell verletzt werden.

Eine Sonderform der Nebenbedingung ist die Verteilung der Verantwortlichkeit (SoD).

Strukturierung des RBAC-Modells

Ein sehr mächtiges Werkzeug des RBAC ist die Strukturierung des Modells. Dabei gibt es verschiedene Strukturierungsmittel: Die Rollen-Rollen-Zuordnung ist eine Methode. In [65] wird ausdrücklich davor gewarnt, das Organigramm einer Firma in eine Rollenhierarchie zu übernehmen. Grund ist die Tatsache, dass die organisatorische Innen- und Außendarstellung sehr unterschiedlich motiviert ist, einer Vielzahl von Zielen gerecht werden muss und in den seltensten Fällen über Arbeitsabläufe und damit verbundene Zugriffsrechte erstellt wurde.

Moffett und Lupu unterscheiden verschiedene Möglichkeiten der Hierarchiebildung:

Generalisierung: Die Generalisierung (Abb. 2.6) ist bekannt als eine "Ist ein"-Hierachie. Sie wird häufig in RBAC-Beispielen verwendet (Ein "wissenschaflicher Angestellter" ist ein "Mitarbeiter" ist ein "Universitätsangehöriger"). Dabei können einige Rollen abstrakt sein; d.h. "Universitätsangehöriger" hat keine direkten Mitglieder, sondern setzt sich aus "Mitarbeitern", "Studierenden" und "Sonstigen Universitätsangehörigen" zusammen.

Aggregation: Die Aggregation (Abb. 2.7) ist als eine "Teil von"-Hierarchie bekannt. Gemäß Moffett und Lupu sollte eine solche Hierarchie über die Relationen "Verantwortlich für" und "Führt aus" definiert werden. Die verantwortliche Einheit kann eine Aufgabe entweder selbst ausführen oder an jemand anderen delegieren. In der Realität ist das oft aber nicht zwingend eine Untereinheit. Eine beliebige Einheit ist diesem Prinzip gemäß über die Menge ihrer Verantwortlichkeiten auf der einen Seite und ihren Ausführungen auf der anderen Seite definiert.

Supervision: Supervision (Abb. 2.8) oder "Leitung und Überwachung" ist ein typisches Modell, wie es in Organigrammen zu finden ist. Einheiten sind hier über ihre Leitung definiert, die in der Regel die Aufsicht und Koordination über die untergeordneten Einheiten und Personen besitzt. Wie bereits erwähnt, werden solche Hierarchien selten im Hinblick auf tatsächliche Rechtevergaben entworfen. Zum Teil hängen Zugehörigkeiten hier von der räumlichen Nähe, der Historie oder auch von persönlichen Gegebenheiten ab. Auf der anderen Seite kann es auf Grund der formalen Verantwortlichkeit sinnvoll sein, die "Vorgesetztenhierarchie" abzubilden.

Schutz vor zu wenigen Rechten

Die Folgen von Rechteüberschreitungen oder von fehlerhafter Autorisierung der falschen Person, d.h. der Zuweisung von zu vielen Rechten an diese Person ist intuitiv verständlich und

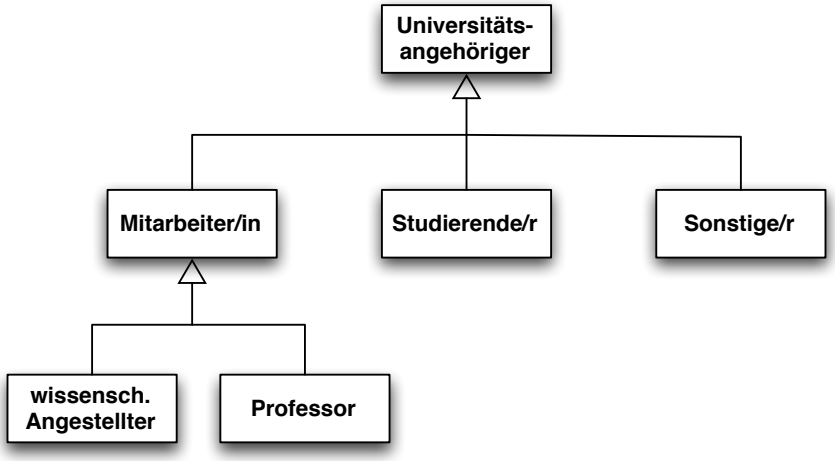

Abbildung 2.6: Klassendiagramm: Beispiel für Generalisierung im Rollenmodell

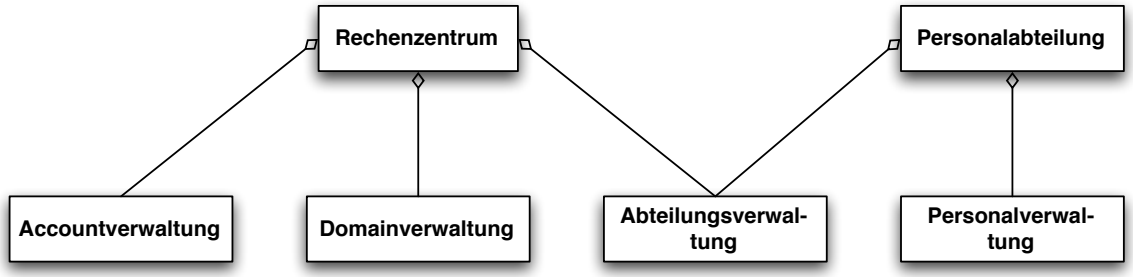

Abbildung 2.7: Klassendiagramm: Beispiel für Aggregation im Rollenmodell

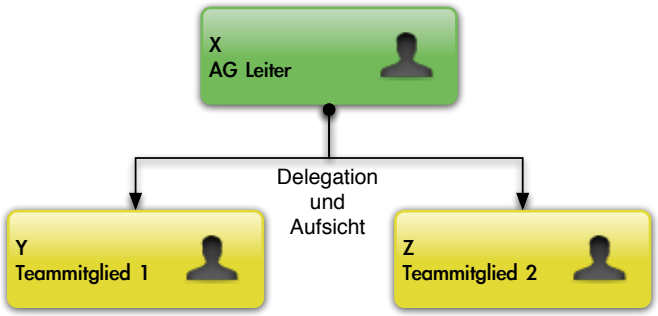

Abbildung 2.8: Klassendiagramm: Beispiel für Supervision im Rollenmodell

wird gerne und ausgiebig diskutiert. Vergessen wird dabei, dass eine Vergabe unnötiger Rechte zunächst nicht zwangsweise zu einer Ausnutzung dieser Rechte und damit zu einem Schaden führen muss. Hingegen kann die Vergabe zu weniger Rechte unmittelbar dazu führen, dass Prozesse in einer Organisation zum Erliegen kommen. Verantwortungsbewusstes Sicherheitsmanagement muss daher nicht nur die Einschränkung von Rechten bedenken, sondern auch die Sicherstellung von hinreichend vielen Rechten. Leicht lassen sich Fälle konstruieren, in denen Angriffe gegen die IT-Infrastruktur abgewendet werden könnten, wenn geschultes Personal mit hinreichend vielen Rechten ausgestattet ist.

Insbesondere für Ausfälle von Personen z.B. durch Krankheit, Dienstreisen, Urlaub usw. müssen Vertreter mit hinreichend vielen Rechten ausgestattet sein oder im Notfall damit spontan ausgestattet werden können. Moffet [65] spricht daher von "Back-up Roles". In diesem Zusammenhang wird vom Prinzip "Passwort im Umschlag" abgeraten, weil hier die Nachvollziehbarkeit (Transparenz) der Aktionen nicht mehr gegeben ist. Später lässt sich nicht mehr nachweisen, wer den Umschlag geöffnet hat und was mit dem Inhalt geschehen ist. Stattdessen wird eine Aufwärtsvererbung von Zugriffsrechten auf genau einer Ebene propagiert. Im Bedarfsfall kann der Vorgesetzte die Blockierung auflösen oder durch Delegation dafür sorgen, dass die Handlungsfähigkeit wieder hergestellt ist.

2.2.7 Implementierungssprachen

Das allgemeine Rollenmodell kann, wie in Abbildung 2.9 als UML-Klassendiagramm dargestellt werden [92].

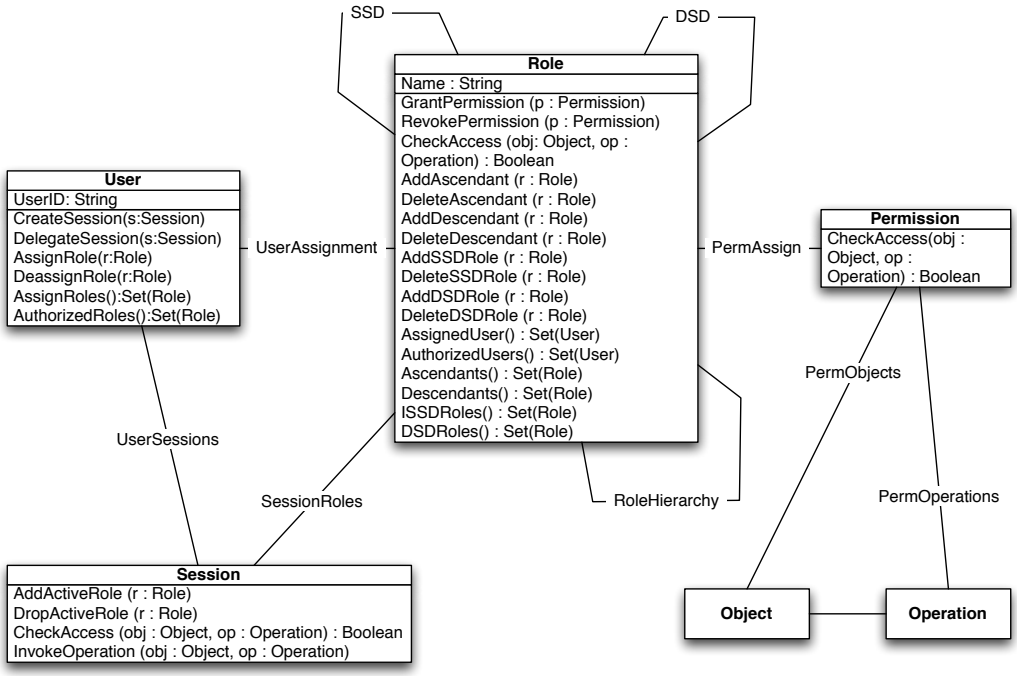

Abbildung 2.9: Klassendiagramm: Allgemeines RBAC-Modell in UML-Repräsentation

Diesem allgemeinen Rollenmodell stehen die konkreten Rollenmodelle entgegen. Dabei handelt es sich aus objektorientierter Sicht um das entsprechende Objektmodell. Benutzer, Rollen

und Rechte werden mit konkreten Daten aus dem Anwendungsbereich des zu schützenden Systems belegt.

Es gibt einige sehr unterschiedliche Ansätze, um ein konkretes RBAC-Modell zu implementieren. Neben anwendungsspezifischen Implementierungen von RBAC-Modellen in Anwendungen, Datenbanken [24] oder auch im Betriebssystem [73, 74], gibt es mehrere allgemeine Möglichkeiten, RBAC-Systeme zu implementieren. Neumann [70] beschreibt eine objektorientierte Skriptsprache (xoTCL), mit der es möglich ist, konkrete RBAC-Modelle zu implementieren. Die TU Berlin entwarf im EU-Projekt MultiPLECX einen Interpreter, mit dem ein konkretes Rollenmodell implementiert werden konnte [30]. Im Anhang von [24] wird ein XML Schema für ein RBAC Modell vorgestellt (siehe auch [12]), mit dem ein RBAC-Modell für eine Organisation aufgebaut werden kann. Üblich sind jedoch Implementierungen auf Basis eines Verzeichnisdienstes [88] oder einer relationalen Datenbank sowie verteilte RBAC Modelle auf Basis von XACML [5].

2.3 Role-Engineering

Auch wenn RBAC mit der Motivation entwickelt wurde, einen intuitiven Zugang zur Zugriffskontrolle zu eröffnen, der dem Alltag in der Geschäftswelt oder auch z.B. in Hochschulen entspricht, stellt die Implementierung eines konkreten Rollenmodells den Modellierer vor diverse Fragestellungen und zu treffende Entwurfsentscheidungen.

Oft wird zwischen "Rollenzuteilung" und "Rollenadministration" unterschieden. Bei der Zuteilung handelt es sich um den Prozess, einer Rolle Mitglieder hinzuzufügen oder Mitglieder aus den Rollen zu entfernen. Die Rollendefinition bleibt dabei unverändert. Demgegenüber steht die Rollenadministration, die auch die Definition der an die Rollen gebundenen Rechte beinhaltet.

Die systematische Ableitung eines konkreten Rollenmodells für einen Anwendungsbereich wird Role-Engineering genannt [15]. Mit wachsendem Einsatz von unternehmensumfassenden RBAC-Systemen in unterschiedlichen Anwendungsfeldern wächst auch der Bedarf an Methoden und deren Beschreibungen zur Modellierung [16].

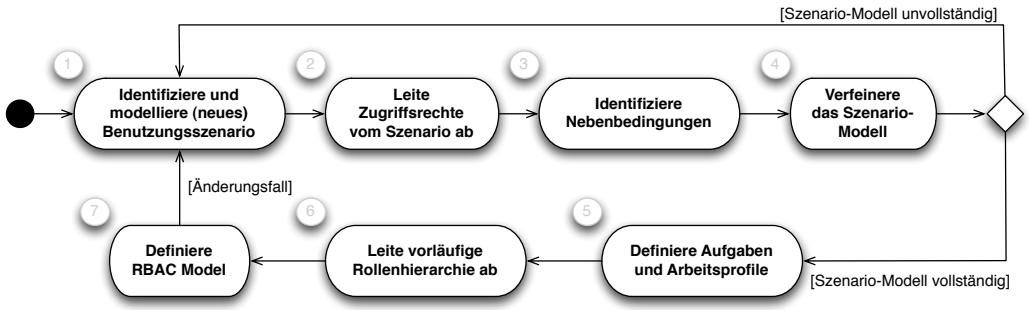

Abbildung 2.10: Zustandsdiagramm: Überblick über das Szenario-basierte Role-Engineering

Neumann und Strembeck entwickelten einen Role-Engineering Prozess, der auf Szenarien basiert [71]. Der Prozess umfasst sieben Phasen und kann sowohl für die Initialisierung, als auch für das Hinzufügen von Szenarien verwendet werden. Abbildung 2.10 zeigt einen Überblick über das Verfahren. Dieses Verfahren wurde von Gao et. al um einen Aspekt-orientierten

Ansatz erweitert [29]. Alternativ oder als Teil der Anforderungsanalyse [24] kann ein zielorientierter Ansatz, wie von Qingfeng He vorgestellt, verwendet werden [40][3].

Aneta Poniszewska-Maranda [77] zeigt einen Ansatz, der auf einer spezifischen Erweiterung des allgemeinen RBAC-Modells beruht, um so den Rollenfindungsprozess zu vereinfachen. So ist es möglich, in einer Rollenhierarchie die Rollen zu spezialisieren, d.h. Rollen einer bestimmten Unterklasse zu erwarten. Das Prinzip von spezialisierten Rollen an definierten Positionen in der Rollenhierarchie benutzt beispielsweise auch die Stanford Universität [24][4]. Das von uns gleichzeitig an der TU Berlin entwickelte Modell baut auf den gleichen Prinzipien auf [49].

2.4 Role-Mining

Einen anderen Weg geht das Role-Mining [60, 86, 95, 96, 66]. Beim Role-Mining werden vorhandene Personen-Rechtezuweisungen analysiert und an Hand dieser Zuordnungen Rollendefinitionen abgeleitet, indem versucht wird, Regelmäßigkeiten mit Hilfe von Clusterbildungen zu finden. Role-Mining wird hauptsächlich zur Initialisierung eines Rollenmodells eingesetzt, indem ein Status Quo der Rechtedefinitionen aus verschiedenen Zugriffskontrollsystemen in ein RBAC-Modell überführt wird. Im Rahmen des umfassenden Autorisierungsmanagement wird eine Initialisierung des Rollenmodells vorausgesetzt, bevor xRE zur Pflege des Modells eingesetzt werden kann. Eine Kombination mit Role-Mining Verfahren ist ebenso denkbar, wie eine Initialisierung mit einem Minimalmodell und der schrittweisen Erweiterung dieses Modells.

2.5 Ausgewählte Software-Engineering Methoden

Abstrahiert man vom Anwendungsgebiet Zugriffskontrolle, so kann man den Role-Engineering Prozess als Modellierungsaufgabe aus dem Software-Engineering auffassen. Der Entwurf eines konkreten Rollenmodells ist vergleichbar mit dem klassischen Softwareentwurf. Aus objektorientierter Sicht geht es beim Role-Engineering darum, eine Anforderungsanalyse durchzuführen und ein Objektmodell gemäß einem vorgegebenen Muster (vergleichbar mit Entwurfsmustern [28, 11]) zu erstellen. Der Themenkomplex Software-Engineering ist bereits gut erörtert. Dass eine Adaption von Methoden und Techniken aus dem Software-Engineering naheliegt, ist auch daran zu erkennen, dass die im Kapitel 2.3 referenzierten Methoden mindestens durch Methoden aus dem Software-Engineering inspiriert wurden (vergl. [57] und [29], [91] und [71] sowie [97] und [40]).

Wie bereits in Abbildung 2.9 gezeigt, kann das allgemeine RBAC-Modell in UML dargestellt werden. Entsprechend kann die UML-Notation auch für ein Objektmodell verwendet werden. In [4] wird gezeigt, wie Nebenbedingungen entsprechend mit der Object Constraint Language (OCL) modelliert werden können.

Allgemeine und konkrete RBAC-Modelle können also mit Modellierungssprachen der Softwaretechnik dargestellt werden und Methoden des Software-Engineerings sind unter definier-

[3] Ferraiolo zitiert hier den Vortrag [38], der jedoch schriftlich nicht verfügbar ist. Herr He war jedoch so freundlich, mir vergleichbares Material zur Verfügung zu stellen.

[4] Auch [64] erörtert das Stanford Modell, gibt jedoch keine expliziten Quellen an. Vermutlich wird hier ebenfalls Bezug auf [24] genommen. Ferraiolo et al. beziehen sich auf ein "internes Papier", das auch bei intensiver Recherche im Original nicht zu finden war.

ten Randbedingungen auf die Modellierung von RBAC-Modellen anwendbar.

2.5.1 eXtreme Programming

Beim eXtreme Programming (XP) [81, 52, 21] handelt es sich um ein Vorgehensmodell zur Softwareentwicklung, bei dem der Programmcode im Mittelpunkt steht und bei dem Ansätze bis ins Extreme getrieben werden. Damit steht XP zum Teil im direkten Widerspruch zum "klassischen Software-Engineering" [27].

XP geht in den Entwicklungsphasen von sehr kurzen Zyklen aus, in denen von der Anforderungsanalyse bis hin zur Implementierung und dem Testen gearbeitet wird. Diese kurzen Zyklen sind notwendig, um bei Entwicklungen, bei denen sich Anforderungen schnell ändern können, rechtzeitig einlenken zu können.

Die wichtigsten Merkmale des XP sind:

Kleine Releases: Ein Release-Zyklus dauert ein bis drei Monate und besteht aus mehreren Iterationen von ein bis drei Wochen Dauer. Iterationen werden in Arbeitspakete von ein bis drei Tagen Dauer zerlegt.

Planungsspiel: Iterationen werden über Planungsspiele gestartet. Hierbei werden Storys entworfen und priorisiert. Anhand der Prioritäten werden die Ressourcen mit der Implementierung der Storys betraut.

Tests: XP verzichtet auf die Erstellung einer Spezifikation. Die Anforderungen werden durch Storys ausgedrückt. Die Entwicklung von Testfällen ersetzt die Erstellung einer Spezifikation. Die Software arbeitet korrekt, wenn das Testprogramm fehlerlos durchlaufen wurde. Die Tests werden dabei zuerst implementiert.

Systemmetapher: Die Metaphern ersetzen im XP die Softwarearchitektur und sollen Kunden wie Entwicklern das System verständlich machen.

Enfacher Entwurf: Bei der Entwicklung soll die einfachste Implementierung erstellt werden, die funktioniert. Zukünftige Entwicklungen sollen dabei nicht bedacht werden.

Refaktorisierung: Bei der Refaktorisierung wird der Code geändert, um ihn zu vereinfachen. Die Semantik soll dabei beibehalten werden, was durch die Tests überprüft werden kann.

Programmierung in Paaren: Je zwei Entwickler führen gemeinsam Entwurf, Programmierung und Tests durch. Durch wechselnde Paare wird das Wissen um den Code verteilt und durch Kontrolle die Qualität verbessert.

Gemeinsames Code-Eigentum: Jedes Entwicklerpaar kann jederzeit an allen Stellen im Code Verbesserungen durchführen. Nach jeder Änderung müssen die Tests neu durchlaufen werden.

Kontinuierliche Code-Integration: Auf einem dedizierten Integrationsrechner werden in kurzen Abständen alle Module integriert und danach getestet. Bei Fehlern müssen die Änderungen zurückgenommen und die Fehler behoben werden.

40-Stunden-Woche: Um jederzeit ein konzentriertes Arbeiten der Paare sicherzustellen, werden geregelte Arbeitszeiten mit Überstunden nur in absoluten Ausnahmefällen gefordert.

Kundenvertreter im Team: Mindestens ein Vertreter der zukünftigen Anwender soll für die Entwickler permanent zur Verfügung stehen. Er entwickelt ferner die Testfälle für die funktionalen Tests.

Programmierrichtlinien: Die Programmierrichtlinien werden gemeinsam von den Entwicklern erarbeitet und sind dann für das gesamte Programmierteam bindend.

2.5.2 Muster

Wie in der klassischen Architektur, so gibt es auch in der Software-Architektur bestimmte Muster (Pattern), die bei der Konstruktion immer wieder in ähnlicher Weise Verwendung finden. In kurzer Folge erschienen zwei Standardwerke zur Pattern-orientierten Software-Architektur: Gamma et al. [28] und Buschmann et al. [11].

Muster stellen eine Möglichkeit dar, Expertenwissen von erfahrenen Programmierern auf direkt nutzbare Weise weiterzugeben und bieten ferner ein Fundament, auf dessen Basis weiterführende Dialoge unter Softwarearchitekten geführt werden können, nachdem abgesteckt ist, von welchen bekannten Mustern ausgegangen wird.

Unter Software-Mustern versteht man z.B. einen Proxy, der den Zugriff auf Objekte mit Hilfe eines Stellvertreterobjektes kontrolliert oder Pipes-and-Filters, die einen Datenstrom immer weitergeben, wobei der Datenstrom in den einzelnen Stationen gefiltert, also geändert, erkannt oder analysiert wird.

Auch oder gerade im Bereich IT-Sicherheit bieten Muster eine Möglichkeit, die Komplexität zu reduzieren oder Rahmen zu schaffen, die es Programmierern ohne Spezialwissen im Bereich der IT-Sicherheit erleichtern, kapitale Sicherheitsfehler beim Entwurf der Software zu vermeiden [89]. Die Verwendung von RBAC wird hierbei als ein Sicherheitsmuster betrachtet. Gemeinsam mit Thomas Gebhardt diskutierte ich in [31] die Definition von Mustern innerhalb der RBAC-Modelle als weiterführenden Schritt.

2.6 Mehrseitige Sicherheit

Mehrseitige Sicherheit beschäftigt sich mit der Durchsetzung von Sicherheitsanforderungen aus mehreren Sichten [80]. Bei heute eingesetzten IT-Systemen sind in der Regel zwei oder mehr Kommunikationspartner involviert. Im Falle einer typischen Webportalanwendung gibt es den Anbieter des Portals und den Nutzer. Beide haben ihre eigenen Sicherheitsanforderungen, die sich ggf. sogar scheinbar widersprechen können. Der Anbieter fordert z.B. aus rechtlichen Gründen, dass der Dienst nur für einen bestimmten Personenkreis (z.B. Mitglieder einer Universität) erbracht wird, wohingegen der Benutzer seine Anonymität (oder zumindest Pseudonymität) fordert.

Der Forschungsbereich mehrseitige Sicherheit befasst sich mit der Umsetzung solcher mehrseitigen Anforderungen und definiert allgemeine Schutzziele sowie Verfahren und Komponenten zur Durchsetzung dieser Schutzziele [67, 69, 68].

Während eines Vortrags am 8. Februar 2008 in der TU Berlin antwortet Prof. Pfitzmann auf die Frage, im Gegensatz wozu denn die "mehrseitige Sicherheit" stehe und ob es Alternativen gäbe: "In einer Demokratie gibt es keine Alternativen zur mehrseitigen Sicherheit." Das Individuum ist in Deutschland durch die Verfassung geschützt. Die Rechte des Individuums sind nicht verhandelbar. Daraus kann die "Sicherheit aller Beteiligten" als feste Anforderung

abgeleitet werden. Das umfasst auch die Erkenntnis, dass der "Bürger" dem Staat, in dem er lebt, nicht in allen Dingen "vertrauen" muss.

Die Zahl der möglichen Angreifer ist auf Grund der Komplexität heutiger EDV-Systeme kaum noch überschaubar. In [80] werden u.a. folgende potentielle Angreifer genannt:

- Außenstehende,
- Benutzer des Systems,
- Betreiber des Systems,
- Wartungsdienst,
- Produzent des Systems,
- Entwerfer des Systems,
- Produzent der für Entwurf und Produktion des Systems verwendeten Hilfsmittel,
- Entwerfer der für Entwurf und Produktion des Systems verwendeten Hilfsmittel,
- ...

Dazu kommen auch die Betreiber der Hilfsmittel und deren Wartungsdienste usw. An jeder Stelle könnte natürlich auch Malware, wie trojanische Pferde oder Viren eine Rolle spielen.

Der Ansatz, jeden Kommunikationspartner vor jedem anderen zu schützen, unterbricht so auch die Kette der durch einen Angreifer kompromittierbaren Systeme. Die allgemeinen Schutzziele werden in [80] folgendermaßen definiert:

Schutzziel Vertraulichkeit (Confidentiality)

c1 Nachrichteninhalte sollen vor allen Instanzen außer dem Kommunikationspartner vertraulich bleiben.

c2 Sender und/oder Empfänger von Nachrichten sollen voreinander anonym bleiben können, und Unbeteiligte (inkl. Netzbetreiber) sollen nicht in der Lage sein, sie zu beobachten.

c3 Weder potentielle Kommunikationspartner noch Unbeteiligte (inkl. Netzbetreiber) sollen ohne Einwilligung den momentanen Ort einer mobilen Teilnehmerstation bzw. des sie benutzenden Teilnehmers ermitteln können.

Schutzziel Integrität (Integrity)

i1 Fälschungen von Nachrichteninhalten (inkl. des Absenders) sollen erkannt werden.

Schutzziel Verfügbarkeit (Availability)

a1 Das Netz ermöglicht Kommunikation zwischen allen Partnern, die dies wünschen (und denen es nicht verboten ist).

Schutzziel Zurechenbarkeit (Accountability)

z1 Gegenüber einem Dritten soll der Empfänger nachweisen können, dass Instanz x die Nachricht y gesendet hat.

z2 Der Absender soll das Absenden einer Nachricht mit korrektem Inhalt beweisen können, möglichst sogar den Empfang der Nachricht.

z3 Niemand kann dem Netzbetreiber Entgelte für erbrachte Dienstleistungen vorenthalten – zumindest erhält der Netzbetreiber bei Dienstinanspruchnahme entsprechende Beweismittel. Umgekehrt kann der Netzbetreiber nur für korrekt erbrachte Dienstleistungen Entgelte fordern.

Neben erweiterten bzw. konkretisierten Schutzzielen werden im Bereich der mehrseitigen Sicherheit auch Technologien zur Erreichung dieser Schutzziele zusammengestellt [76]. In dieser Arbeit wird dabei vornehmlich von den Chaum'chen Mixen Gebrauch gemacht [13]. Das Prinzip der Mix-Technologie kann wie folgt beschrieben werden:

Eine elektronische Nachricht wird durch mehrere Stationen, sog. Mixen geschickt. Bei jeder Station wird der Absender der Nachricht umkodiert, so dass nur in der Mix-Station der ursprüngliche Absender wieder rekonstruiert werden kann. Am Ende einer Mix-Kaskade ist es nicht möglich, den Absender der Nachricht zu ermitteln. Die Antwort wird an die letzte Station der Mix-Kaskade gesendet, die durch Rückkodierung die vorletzte Absenderadresse errechnen kann. Die Nachricht durchläuft die Kaskade bis zum Ausgangspunkt, wo wiederum der ursprüngliche Adressat ermittelt werden kann.

Je mehr Nachrichten durch die Mix-Kaskaden laufen, desto schwieriger ist es auch durch statistische Verfahren Nachrichten einem Absender und einem Empfänger zuzuordnen. Anonymisierungsdienste, wie JonDonym (`https://www.jondos.de/`) bieten auch Mixen in verschiedenen Ländern an, um damit die Zensur oder die Protokollierung auf Basis eines Generalverdachts zu umgehen. Der Anonymisierungsdienst Tor (`http://www.tor.de/`) nutzt die Rechner der Tor-Teilnehmer, um verschlüsselte Informationen über unterschiedliche Stationen zu verschleiern. Tor ermöglicht die Nutzung unterschiedlicher Internetprotokolle und ist nicht auf das WWW beschränkt.

2.7 Sicherheitsfaktor Mensch

Wie bei praktisch allen sicherheitsrelevanten Methoden ist die Dokumentation, die Schulung und der Support durch Experten von größter Wichtigkeit. Eine, wenn nicht die entscheidende Angriffsfläche eines Sicherheitssystems ist der Mensch und dessen Beeinflussbarkeit. Wie jedes Sicherheitssystem, ist auch RBAC gegen Social-Engineering anfällig [87]. Dem kann nur mit einem möglichst hohen Wissensstand (Sensibilisierung) der Mitarbeiterinnen und Mitarbeiter begegnet werden. Dazu gehört selbstverständlich auch eine geeignete Benutzerführung der verwendeten Software, ein umfassendes Lehr-, Auffrischungs- und Weiterbildungsangebot sowie ein kompetenter Support, der im Zweifelsfall Auskunft geben kann.

Kapitel 3

Umfassendes Autorisierungsmanagement

> Eureka Maru: Warning, specified proximity exceeds safety margin.
> Beka: Override safety protocols, authorization code: 'Shut up and do what I tell you!'
> Eureka Maru: Authorization accepted.
>
> Andromeda (TV series)

Im Folgenden wird ein normatives Modell für ein umfassendes Autorisierungsmangement vorgestellt. Der Begriff "umfassend" charakterisiert die Lösung als ein System, das die Autorisierung für viele unterschiedliche Teilsysteme verwaltet. Dabei sollten die *wesentlichen* Prozesse der Organisation über das Autorisierungsmanagement gesteuert werden, damit von einer "umfassenden Lösung" gesprochen werden kann. Die Anzahl, die Art und der Umfang der Integration in das Autorisierungsmanagement ist dabei stark von der jeweiligen Organisation abhängig. Die zweite Dimension der "umfassenden Lösung" betrifft die Zahl der Nutzer, die durch das System versorgt werden. Ein umfassendes System sollte so viele Benutzer wie möglich versorgen, kann also ggf. sowohl Mitarbeiter, externe Benutzer (wie z.B. Zulieferer etc.) bis hin zu Kunden berücksichtigen.

Aus dem Modell kann ein Rahmenwerk für ein Autorisierungssystem implementiert werden, das Autorisierungsinformationen mit unterschiedlichen Anwendungen koppelt. Durch diese Kopplung kann ein umfassendes, widerspruchsfreies Zugriffsmodell implementiert werden. Durch die Einführung eines solchen Systems werden ferner Synergien geschaffen und der Verwaltungsaufwand wird reduziert.

Mit dem im Folgenden vorgestellten "umfassenden Autorisierungssystem" ist es möglich, den Anforderungen aus den verschiedenen Szenarien gerecht zu werden. Dabei ist die Lösung als *eine* mögliche Variante zu verstehen. Der Themenkomplex "umfassende Autorisierungssysteme" wird an Hand eines Beispiels beleuchtet. Der Kern dieser Arbeit kann mit Hilfe der Abb. 3.1 umschrieben werden.

Die Grafik zeigt, dass das umfassende Autorisierungsmanagement Aspekte auf unterschiedlichen Anwendungsschichten umfasst. Das System setzt auf der organisatorischen Ebene auf. Die organisatorischen Aspekte müssen mit Hilfe geeigneter Architekturen umgesetzt werden. Die Architekturen basieren auf adäquaten Technologien.

Der Schwerpunkt dieser Arbeit liegt auf den rot markierten Aspekten, d.h. der Modellierung der verteilten, dezentralen Administration der Zugriffsverwaltung, dem Ansatz, das vom Softwareengineering bekannte Prinzip der Muster auch für RBAC Systeme anzuwenden und dem Einsatz von Techniken der mehrseitigen Sicherheit. Diese drei Aspekte sind Weiterentwicklungen verschiedener bereits bekannter Arbeiten im Bereich IT Sicherheit und

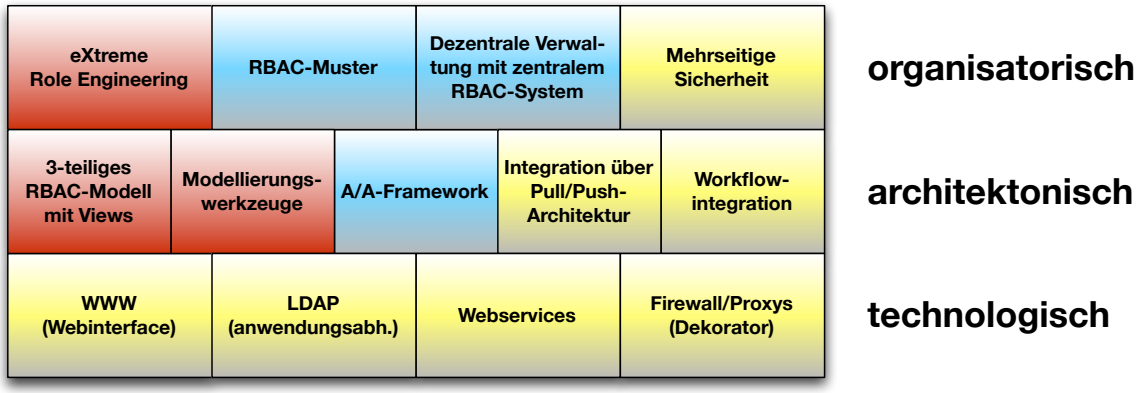

Abbildung 3.1: Überblick über die Aspekte des umfassenden Autorisierungsmanagements

Softwareentwicklung. Dem nach dem Vorbild des eXtreme Programming neu entwickelten Role-Engineering Verfahren "eXtreme Role-Engineering" ist das Kapitel 4 gewidmet. Details zur technischen Realisierung sind im Erfahrungsbericht in Kapitel 5 erörtert. Die blau unterlegten Aspekte wurden in Bezug auf die Anwendungsgebiete verbessert oder auf neue Bereiche angewandt. Die gelben Aspekte werden gemäß dem Stand der Technik benutzt.

3.1 Anforderungen an das umfassende Autorisierungsmanagement

Diese Arbeit berücksichtigt Systemanforderungen, die aus sehr unterschiedlichen Projektumfeldern zusammengetragen wurden. Hierzu gehören Anforderungen, die aus dem Business-to-Business (B2B) Kontext kommen, sowie Anforderungen aus dem universitären Umfeld. Die gemeinsamen Anforderungen aus so unterschiedlichen Domänen legt die Vermutung nahe, dass ähnliche Problemstellungen in noch weiteren Bereichen existieren werden.

3.1.1 Die Anwendungsfälle: B2B-Plattform und Universität

Die B2B-Anforderungen sind hauptsächlich dem EU-Projekt MultiPLECX entnommen, bei dem das RBAC-System zur Zugriffssteuerung eines Marktplatzes und eines Kundenportals genutzt wurde. In beiden Fällen sollten die jeweiligen Geschäftspartner die Zugriffe ihrer eigenen Mitarbeiter autonom verwalten können. In beiden Fällen war darauf zu achten, dass die anderen Geschäftspartner keinen Einblick in die Zugriffsmodelle der jeweils anderen erhalten, um keine Schlussfolgerungen in Bezug auf die internen Strukturen ziehen zu können.

Bei der praktischen Einführung eines RBAC Systems an der TU Berlin [41] stellten sich sehr ähnliche Anforderungen. So existieren an der Universität verschiedene Einheiten, die große unabhängige Handlungsbefugnisse besitzen. Hier sind es rechtliche Rahmenbedingungen, wie der Datenschutz, die eine Einschränkung der Sichtbarkeit nötig machen.

Im Kontext der sich wandelnden Rahmenbedingungen für die Universitäten wie z.B. durch den Bologna-Prozess [19], wäre ein föderatives System sinnvoll, das von vielen, wenn nicht mindestens allen deutschen Hochschulen genutzt werden könnte. In einem solch föderativen System zeigen sich z.T. ähnliche, in weiten Teilen jedoch die selben Anforderungen.

3.1.2 Gemeinsame Anforderungen

Im Folgenden werden die unterschiedlichen Anwendungsfälle nicht weiter differenziert. Auch wenn sich die Motivation unterscheidet, ergeben sich Anforderungen, die sich mindestens die drei genannten Szenarien teilen.

Verwaltung der Repräsentation der eigenen Organisation / Einheit

Umfasst das Rollenmodell die Repräsentation von verschiedenen gemeinsam agierenden Organisationen oder Einheiten mit weitreichender Autorität bzw. Handlungsbefugnis, so muss die Einheit in die Lage versetzt werden, die Administration ihrer eigenen Repräsentation im Modell durchzuführen. Wie weit die jeweiligen Befugnisse reichen, ist dabei wiederum abhängig vom jeweiligen Szenario. Die Einheitenverwaltung kann beispielsweise die folgenden Funktionen umfassen:

- Zuteilung von Aufgaben an Mitarbeiter der Einheit und damit Zuweisung von Rollenmitgliedschaften.
- Definition eigener Geschäftsrollen und damit Zuteilung von Rechten an Benutzer.
- Spezifizierung von Vertretungen.
- Delegation von Aufgaben und damit Rollen und Rechten.

Eine weitere mögliche Anforderung ist die Vertraulichkeit der Strukturinformationen. In den Rollendefinitionen und Rollenmitgliedschaften sind wesentliche Informationen über die Organisationen oder Einheiten enthalten, die diese nicht preisgeben wollen oder müssen. Es ist auch denkbar, dass solche Informationen den Datenschutz berühren. Ferner beugt eine Begrenzung der Sichtbarkeit der Informationen auch verschiedenen Angriffen gegen die Einheit vor (z.B. bei der Vorbereitung von Social Engineering Angriffen).

Trennung von Anwendungs- und Geschäftslogik

Die Trennung der Verantwortlichkeiten ist im Bereich der Anwendungsbetreibung naheliegend. Üblicherweise gibt es bestimmte Einheiten, die für den Betrieb einer Anwendung zuständig sind, und in der Regel ist eine dieser Einheiten bestenfalls einer der Nutzer der Anwendung. Die strikte Trennung von Administration einer Ressource und deren Nutzung ist seit Jahren gängige Praxis in der IT-Sicherheit.

Da Anwendungsbetreiber üblicherweise in weiten Teilen für die Betriebssicherheit der Anwendung verantwortlich sind, fordern sie zurecht eine Kontrolle über die Nutzer ihrer Anwendung. Auf der anderen Seite steht ihnen jedoch nur selten eine personelle Entscheidung zu. Die Schnittstelle bietet das zur Verfügungstellen von Rechten an eine Organisation oder Einheit oder einer Klasse von Einheiten (z.B. alle Verkäuferfirmen eines Marktplatzes oder alle Fakultäten einer Hochschule).

Zu den Aufgaben der Anwendungsverwaltung gehören:

- Definition von Rechten für die Nutzung einzelner Funktionen der Anwendung.
- Zusammenfassen von Rechtegruppen zu Anwendungsrollen.

- Bereitstellen von Anwendungsrollen für Einheiten oder Einheitenklassen.

Hinter der Definition von Anwendungsrollen steckt die Möglichkeit für den Anwendungsbetreiber, die Sicht auf das Rollenmodell aus Anwendungssicht zu modellieren. Eine Anwendung unterscheidet z.B. nur zwischen Kunden und Anbieter, eine andere muss dagegen Honorarprofessoren von Juniorprofessoren oder Emeriti unterscheiden.

Aus softwaretechnischer Sicht entkoppelt die Trennung von Anwendungs- und Geschäftslogik diese beiden Modellaspekte voneinander. Neben der wohl definierten Trennung der Verantwortlichkeiten gewinnt man hierdurch eine Stabilität gegen Änderungen in Anwendungen. Zum einen können die geänderten Funktionen in das bestehende Rechte- oder Anwendungsrollenmodell eingearbeitet werden, zum anderen betreffen Änderungen an den zur Verfügung gestellten Anwendungsrollen nur die jeweilige Anwendung. Eine Beeinflussung der Geschäftslogik oder Implikationen in Bezug auf andere Anwendungen kann so ausgeschlossen werden.

Dynamische Strukturänderungen

Das Autorisierungssystem muss robust gegen strukturelle Änderungen sein. Ob im elektronischen Marktplatz Verkäufer hinzukommen oder entfallen oder ob in der Universität Fachbereiche zusammengelegt werden oder neue Professuren geschaffen werden, Einheiten können hinzukommen, sich ändern oder entfernt werden. Der Aufwand, solche Änderungen in das Autorisierungsmodell zu übernehmen, sollte möglichst gering sein. Das Modell ist so zu gestalten, dass es gegen diese Arten von Änderungen möglichst robust ist, also möglichst wenig Implikationen bestehen.

Dilemma zwischen zentralen und dezentralen Aspekten

In den beschriebenen Szenarien besteht die Forderung nach gemeinsamen Ressourcen, die vom Autorisierungssystem zu schützen sind, gemeinsamen Anwendungen, die den Zugriff auf die Ressourcen ermöglichen aber einer dezentralen Verwaltung der Rechte auf diese Anwendungen bzw. Ressourcen.

Das Autorisierungsmodell muss die verschiedenen Sichtweisen berücksichtigen, jedoch eine gemeinsame Sicherheitspolitik umsetzen.

Schutz des RBAC Modells

Ein besonderes Augenmerk bei der Umsetzung des zentralen Autorisierungssystems muss auf dem Schutz des Modells liegen[1]. Dies umfasst alle Aspekte des Datenschutzes. Die folgenden Anforderungen spielen dabei eine besondere Rolle:

Fehlbedienungen: Die dezentrale Administration hat zur Folge, dass viele Administratoren mit unterschiedlicher Qualifikation am Modell arbeiten. Das Autorisierungssystem muss daher passive, wie aktive Sicherheiten gegen Fehlbedienungen bieten. Dazu gehört die Begrenzung von Fehlern auf die jeweiligen Einheiten, eine möglichst umfassende Umsetzung der Sicherheitspolitik, die gravierende Fehler ausschließen soll aber auch eine Benutzerführung, die aktiv Fehler vermeiden soll.

[1] Eine allgemeine Zusammenfassung von Schutzmechanismen in RBAC-Systemen ist in Kapitel 2.2.6 zusammengestellt.

Softwarefehler: Ein umfassendes System ist potentiell anfällig gegen Entwurfs- und Programmierungsfehler. Die Architektur des Systems muss diesen Umstand berücksichtigen.

Angriffe: Das Autorisierungssystem muss Angriffe sowohl von innen, wie auch von außen abwehren können. Dabei ist zu berücksichtigen, dass die Angriffe nicht nur von jeder Person, sondern auch auf jeder Architekturebene erfolgen können.

Nutzung vorhandener Ressourcen

Häufig sehen IDM Systeme vor, dass das IDM die bisherigen Systeme zur Personenverwaltung ersetzt, parallel dazu existiert oder synchronisiert. Alternativ kann das IDM jedoch auch an die existierenden Systeme angeschlossen werden, um diese zu verknüpfen und die Informationen für das Autorisierungsmangement zu nutzen. Die vorhandenen Daten können so unberührt bleiben. Eine doppelte Datenhaltung wird vermieden.

Das System muss ferner in der Lage sein, weitere Quellen aufzunehmen, um beispielsweise Benutzerkreise zu erweitern, die mit den bestehenden Systemen gar nicht verwaltet werden können.

Typische Ressourcen, die ggf. existieren und genutzt werden können sind:

- Personaldaten (in Auszügen)
- Studierendendaten bzw. Kundendaten
- Kostenstellenverzeichnis bzw. Organigramm

Auch die vorhandenen Softwaresysteme sollen weiter zum Einsatz kommen. So ist es an vielen Stellen nicht denkbar, eine Software in Hinblick auf das umfassende Rollenmodell neu zu entwickeln oder entsprechend anzupassen. Es muss möglich sein, Softwaresysteme mit verschiedener Integrationstiefe in das Konzept einzubetten. Bei einigen Systemen ersetzt das Rollensystem z.B. den Anmeldevorgang und stellt ein Single Sign On zur Verfügung. Bei einem anderen System entscheidet die Rollenmitgliedschaft über die nutzbaren Module, bei anderen Programmen muss die Logik des umfassenden Autorisierungsmanagements auf die Rollenlogik des zu integrierenden Programms angepasst werden. Selbstverständlich muss es auch Programmierschnittstellen geben, um Anwendungen erstellen zu können, die speziell auf das umfassende Autorisierungssystem zugeschnitten sind.

Gesetzeskonformität und Einführungsblockaden

Mit der Einführung des umfassenden Autorisierungsmanagements ist geltendes Recht einzuhalten. Das betrifft insbesondere den Datenschutz aber auch andere Regelungen, wie die Mitbestimmung der Interessensvertretungen.

Ein System, dass sich an den Konzepten der mehrseitigen Sicherheit orientiert und die informationelle Selbstbestimmung fördert sowie die Handlungsautonomie bewahrt, ist zum Abbau von Einführungsblockaden gut geeignet. Insbesondere weil es sich bei der Einführung des Systems um eine umfassende Maßnahme handelt, ist die frühzeitige Einbeziehung aller betroffener Parteien entscheidend. Die Verbesserung der Transparenz von Vorgängen und der damit verbundenen Rechte kommt am Ende allen Parteien zugute. Mit Schwierigkeiten ist vor allem an den Stellen zu rechnen, an denen bislang Einigkeit nur durch Verschleierung erzielt wurde.

Aus arbeitswissenschaftlicher Sicht ist die Erweiterung der Entscheidungsbefugnisse und der Einflussnahme auf die Arbeitsabläufe grundsätzlich positiv zu bewerten [20, 98]. In Bezug auf die Produktivität der Organisation aber auch der Vermeidung von sog. Regulationshindernissen [62] muss eine hohe Verfügbarkeit des Autorisierungssystems erreicht werden.

Kann für alle betroffenen Parteien eine objektive Verbesserung der Arbeitssituation und damit auch der Produktivität erreicht werden und können auf der anderen Seite die Zuständigkeiten und die Verteilung der Verantwortung (so fern sinnvoll) beibehalten werden, so können Einführungsblockaden leicht überwunden werden.

3.2 Lösungsansätze und Überblick

Der folgende Abschnitt fasst verschiedene Lösungsansätze zur Erfüllung der benannten Anforderungen zusammen und präsentiert weiterentwickelte und alternative Ansätze, die schließlich in den folgenden Abschnitten ausgearbeitet werden.

Verteilte Administration

Ferraiolo widmet in [24] dem Thema "Role-Based Administration of RBAC" ein ganzes Kapitel. Ein sehr früher Ansatz hierzu stammt von Sandhu [84]. ARBAC definiert ein zweigeteiltes RBAC-Modell. Der eine Teil definiert den Zugriff auf Ressourcen, der andere den Zugriff auf das RBAC-Modell. Alternative Modelle verzichten auf eine Trennung von administrativen Rollen und Zugriffsrollen. Der Schwerpunkt dieser Modelle liegt auf der Definition des Gültigkeitsbereichs, also der Frage, welche Rollen administriert werden dürfen. Diese Konzepte sind sehr flexibel. Die Definition des Gültigkeitsbereichs ist jedoch komplex und somit fehleranfällig.

An einer anderen Stelle im RBAC-Modell setzen Kumar et al. [55] an. Sie fassen die Objekte, also die zu schützenden Ressourcen zu Objektklassen zusammen und definieren demnach den Zugriff nicht auf einzelne Objekte, sondern auf Objektklassen. Über diesen Weg lassen sich ähnliche Anwendungsfälle definieren, wie über die o.g. Ansätze. Der Unterschied liegt hier darin, dass sich über die sog. Kontexte ebenfalls Gültigkeitsbereiche definieren lassen, diese jedoch nicht über die Vergabe der Rollen gesteuert werden, sondern über die Ressourcen, auf die der Zugriff gewährt wird.

Alternativ zu diesen Konzepten kann die Modelltrennung auch zwischen Rollenmodell und Strukturmodell (Organigramm) erfolgen. Diese Alternative bietet verschiedene Vorteile:

- Die Verwaltung des Strukturmodells kann unabhängig vom Rollenmodell erfolgen. Das verringert die Komplexität und ermöglicht eine Verteilung von Verantwortlichkeiten.
- Über die Zuweisung von Ressourcen und Methoden zu Struktureinheiten kann transparent gesteuert werden, welche Rechte maximal durch eine Einheit verwaltet werden können. Dieses Vorgehen entspricht auch gängiger Praxis in Wirtschaft und Verwaltung.
- Bei Änderungen in der Struktur müssen die Gültigkeitsbereiche nicht neu definiert werden. Sie ergeben sich jeweils aus Struktur und Typisierung der Einheit sowie der für die Typen definierten Nebenbedingungen. Dabei beschreiben die Nebenbedingungen jeweils nur, welche Rollen durch die Administratoren verwaltet werden dürfen. Die Zuteilung von Ressourcen findet unabhängig davon statt.

- Das Rollenmodell kann den Benutzern in Form des Organigramms präsentiert werden. Diese Sichtweise ist den Benutzern meist bekannt. Rollenstrukturen unterscheiden sich in der Regel stark von der Außenrepräsentation, da eine Bündelung besonderer Rechte u.a. auch bei Helpdeskmitarbeitern und IT-Technikern auftritt und ggf. eine Führungsperson nur minimale Rechte benötigt, da sie auf einer Ebene agiert, die von der IT nicht durchdrungen ist oder die zumindest nicht vom Rollensystem verwaltet wird.

Die ersten Modelle, die auf diese Struktur hinführen, werden bereits in [45] beschrieben. Mit der Entwicklung des TUBIS-Systems wurde das Konzept schließlich konsequent durchgesetzt.

Entkopplung der Anwendungsschicht

Nach dem Vorbild der Trennung von Struktur und Rollen wird auch die Anwendungsschicht vom restlichen Modell entkoppelt. Die Anwendungsbetreiber können Rechte und Objekte für ihre Anwendungen definieren und zu Anwendungsrollen zusammenfassen. Diese Anwendungsrollen werden dann den Strukturtypen zugeordnet.

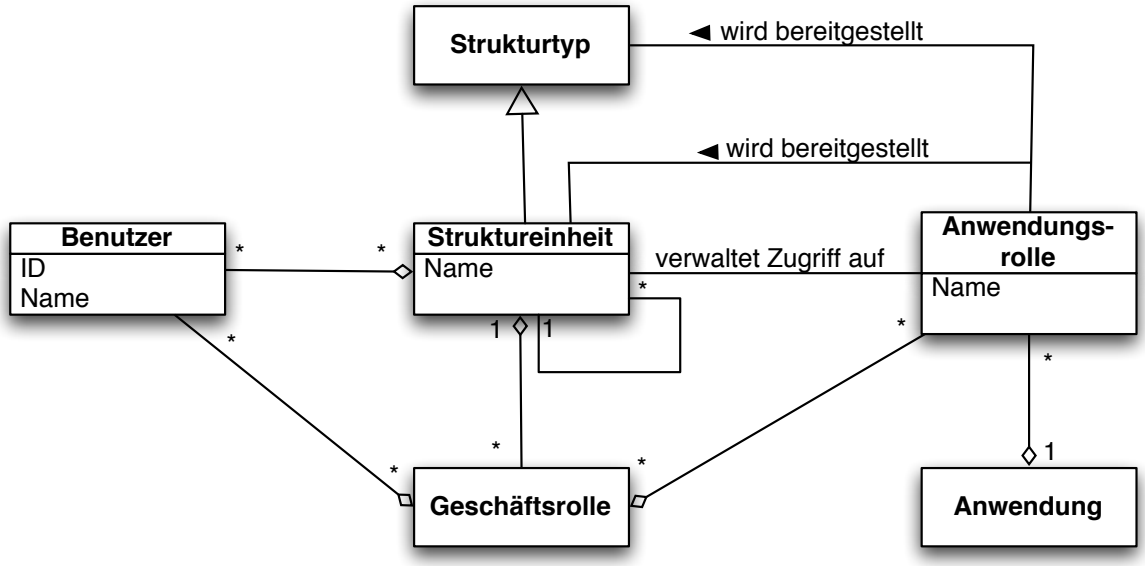

Abbildung 3.2: Klassendiagramm: Entkopplung der Struktureinheiten und der Anwendungen

Einen sehr ähnlichen Ansatz findet man beim Stanford Modell [24]. Dieser, sowie andere sehr ähnliche Modellierungsansätze basieren darauf, die Rollenhierarchien dazu zu nutzen, ein Schichtenmodell mit Rollen verschiedener abgeleiteter Klassen aufzubauen. Im Folgenden gehen wir davon aus, dass Anwendungsrollen nur jeweils Geschäftsrollen zugewiesen werden können. Nur Geschäftsrollen können Mitglieder besitzen. Nur Anwendungsrollen besitzen Rechte für die Nutzung von Methoden auf Objekten.

In der Praxis werden der Anwendung vom Autorisierungssystem meist nur Anwendungsrollen übergeben. Die Anwendung selbst übernimmt dann die Prüfung der Berechtigung an Hand der übergebenen Rolle. Für die Entscheidungen können weitere Daten (Separation-of-Duty Data) relevant sein. Fordert ein Abteilungsleiter beispielsweise Finanzdaten seiner Abteilung an, so ist für die Finanzanwendung nicht nur relevant, dass es sich hierbei um ein Mitglied der

Rolle "Abteilungsleiter" handelt, sondern auch für welche Abteilung die Person diese Rolle inne hat.

Lösungsansätze für den Schutz des Systems

Als Zugriffskontrollsystem ist RBAC von Hause aus so entworfen, dass eine Reihe von Sicherheitsmechanismen zusammenwirken, um fehlerhafte Zugriffsentscheidungen zu vermeiden (siehe Abschnitt 2.2.6). Das Risiko steigt mit dem durch das Rollensystem geschützten Zugriff auf sich selbst. Ein Fehler an dieser Stelle birgt die Gefahr, dass sich ein Benutzer selbst mehr Rechte zuteilen kann und somit eine so genannte Privilegieneskalation möglich ist.

Neben den in Kapitel 2 ausgeführten Instrumenten zur Risikominimierung ist auch die beschriebene Modelltrennung ein geeignetes Werkzeug zum Schutz des RBAC-Systems und der durch das System geschützten Ressourcen. Die Struktur kann durch die verschiedenen Sichten von einer anderer Stelle verwaltet werden, als die Rollen. Werden durch die Anwendungsverwaltung Ressourcen und Zugriffe den jeweiligen Einheiten zugeteilt, so ist eine fehlerhafte Modellierung auf die jeweilige Struktureinheit und ggf. der Untereinheiten beschränkt. Aus diesem Grund ist es ratsam, die Verwaltung von Untereinheiten auf solche zu beschränken, die keinen eigenen Verwalter besitzen. Die Anwendungsverwalter selbst sind lediglich in der Lage, Ressourcen und Rechte für ihre eigene Anwendung zu vergeben. Sie haben keinen Zugriff auf die Rollenzuteilung. So ermöglicht die Trennung des Modells in verschiedene Modellsichten eine weitere Teilung der Verantwortlichkeiten.

Die Teilung der Verantwortlichkeiten ist eine bekannte Methodik der mehrseitigen Sicherheit. Die Anwendung weiterer Werkzeuge der mehrseitigen Sicherheit verbessert den Schutz des gesamten Systems zusätzlich. Anwendung finden Datensparsamkeit, Einschränkungen der Sichtbarkeit sowie die nicht Verkettbarkeit von Informationen.

Nutzung vorhandener Ressourcen

Die Nutzung vorhandener Ressourcen ist ein entscheidender Kostenfaktor, stellt jedoch auch einen Bestandteil der Datensparsamkeit dar. In der Regel existieren bereits Personal-, Kunden- oder im Falle einer Universität Studierendendatenbanken. Statt Daten aus diesen Datenbanken in das Rollensystem zu kopieren[2], kann direkt über geeignete Filter auf diese Datenquellen zugegriffen werden. Ein weiterer Vorteil besteht in der Konsistenz und Aktualität der referenzierten Daten. Jede Änderung in der Primärdatenbank wirkt sich sofort auch im Zugriffssystem aus. Nachteil dieser Methode sind die erhöhten Anforderungen an die Verfügbarkeit der Datenquellen. Ein Ausfall der Personaldatenbank wirkt sich ggf. negativ auf die Verfügbarkeit einer Reihe von Diensten aus.

Eine andere nutzbare Ressource sind vorhandene Anwendungen. Die Integration von bestehenden Anwendungen in das Autorisierungssystem stellt eine typische Anforderung dar. Aber auch bei Neuanschaffungen von Anwendungen wird eine hohe Flexibilität bei der Integration gefordert. Bei web-basierten Anwendungen bietet sich die Verwendung von Proxy-Servern mit Filtereigenschaften an. Die Proxy-Server sind ferner in der Lage, Anfragen mit weiteren Informationen aus dem Autorisierungssystem anzureichern (Dekorator-Muster).

[2]Das RBAC-System wird hier als Teil des IDM gesehen.

Überblick

Die beschriebenen Anforderungen lassen sich durch ein RBAC-System erfüllen. Dabei sind drei Facetten von besonderer Bedeutung:

1. Die dezentrale Verwaltung des zentralen RBAC-Systems.
2. Die Anwendung von aus dem Software Engineering bekannten Techniken zur Qualitätssicherung des Modells.[3]
3. Anwendung von Methoden der mehrseitigen Sicherheit zum Schutz des Modells und zur Umsetzung von rechtlichen Anforderungen.

Ein Zentraler Punkt ist die softwaregestützte Modellierung von Teilmodellen (hier der Geschäftsrollen). Dieser wird im Kapitel 4 ausführlich diskutiert.

3.3 Dezentrale Verwaltung mit zentralem RBAC-System

Das für das umfassende Autorisierungsmanagement entworfene RBAC-Modell realisiert eine eigene Rollenverwaltung für jede Untereinheit. Je nach Anwendungsfall können Untereinheiten eigene Firmen sein (wie im Marktplatzszenario) oder Fachgebiete einer Universität. Grundgerüst des Modells bildet ein Organigramm oder Strukturdiagramm. An der TU Berlin wurde hierfür das Kostenstellenverzeichnis herangezogen, da es die vollständigste Abbildung der Einheiten der Universität darstellte. Personen und Geschäftsrollen werden jeweils einer Einheit im Strukturdiagramm zugeordnet. Dabei können Personen verschiedene Positionen in verschiedenen Untereinheiten besitzen. So gibt es z.B. häufig Sekretärinnen, die jeweils auf einer halben Stelle für zwei Fachgebiete arbeiten. Die so genannten Geschäftsrollen hingegen sind jeweils einer Einheit zugeordnet. Jede Einheit bekommt die für ihre Aufgaben nötigen Ressourcen und Rechte zugewiesen, die sie wiederum über die ihr zugewiesenen Geschäftsrollen an Personen vergeben kann. Dabei müssen Personen nicht zwangsläufig der eigenen Einheit zugeordnet sein.

Es existieren unterschiedliche Ansätze für die dezentrale Verwaltung von RBAC-Systemen [84, 17, 18, 53]. Das hier vorgestellte System legt seinen Schwerpunkt auf die konsequente Verteilung von Verantwortlichkeiten bei der Administration des RBAC-Systems und auf die Möglichkeiten zur Integration in bestehende Infrastrukturen. Die Besonderheiten der Lösung bestehen in der physikalischen Verteilung des Modells sowie der logischen Trennung von Struktur und Rollen. Im Rahmen der Anwendung von aus dem Software-Engineering bekannten Methoden wurden konsequent Sichten (Viewpoints) über eine Model-View-Controller-Architektur realisiert.

3.3.1 Zuständigkeiten und Verantwortungen

Viele IDM bzw. RBAC-Systeme setzen eine Zentralisierung der Datenbestände bis hin zu einer Migration aller Personendaten in das zu etablierende System voraus. Entwurfsziel des hier vorgestellten Systems war eine möglichst hohe Flexibilität bei der Verteilung der Zuständigkeiten und Verantwortungen zu erreichen. Damit existiert auf der organisatorischen Seite die

[3] Zum Einsatz kommen Muster und Sichten sowie eXtreme Programming Ideen bei der Unterstützung der Administratoren bei der Modellierung.

Option, effiziente und etablierte Prozesse zu übernehmen und das Autorisierungsmanagement darin einzubetten und auf der anderen Seite umzustrukturieren, wo im Prozessablauf Mängel vorliegen. Dabei stehen jeweils sowohl die Zentralisierung, wie auch die gezielte Dezentralisierung als Möglichkeiten zur Verfügung. Diese Optionen können Reibungsverluste bei der Umsetzung eines umfassenden Autorisierungsmanagements reduzieren und als technische Voraussetzung für ein solches Vorhaben wirken.

Das Rollenmodell kann aufgeteilt und an unterschiedlichen Stellen gespeichert werden.

Teilidentitäten: Die Personenattribute der vom System verwalteten Benutzer müssen nicht im RBAC-Modell gespeichert werden. Sie können verteilt vorliegen und von einem Metaverzeichnis bei Bedarf angefordert werden. Dabei ist sowohl eine horizontale, wie auch eine vertikale Trennung sowie Kombinationen hieraus möglich. Mitarbeiter-Daten können aus einer anderen Datenbank bezogen werden, Kundendaten und technische Informationen aus anderen Datenquellen.

Geschäftslogik: Unter der Geschäftslogik sind im Sinne des RBAC-Systems die Menge von Rollen zu verstehen, die die Person–Aufgabenzuordnung widerspiegeln. Diese Rollen werden im Folgenden Geschäftsrollen genannt. Die Geschäftsrollen können dezentral von den jeweiligen Untereinheiten einer Organisation (organisational units) verwaltet werden. Die zu Grunde liegenden Daten hierfür können zentral oder auf die Einheiten verteilt gehalten werden. Grundsätzlich besteht keine Notwendigkeit für eine Untereinheit auf die Rollenmitgliedschaften oder Rollendefinitionen einer anderen Einheit zuzugreifen, wie im Folgenden noch dargestellt wird.

Strukturlogik: In RBAC-Systemen wird die Struktur des Systems meist über Rollenhierarchien abgebildet. Die Strukturierung der Organisation kann jedoch auch als eigenständige Aufgabe gesehen werden. Dieser Ansatz ermöglicht zum einen auch hier eine Teilung von Verantwortlichkeiten bis hin zur Option der dezentralen Datenhaltung, zum anderen vereinfacht er die Logik zur dezentralen Verwaltung. Die Verwaltung der Strukturlogik selbst kann wiederum aufgeteilt werden. Dabei werden die Berechtigungen auf die Verwaltung der Struktur aus dem RBAC-System oder unabhängig davon verwaltet.

Anwendungslogik: Wir setzen voraus, dass für jede in das System integrierte Anwendung mindestens eine verantwortliche Person existiert. Ferner wird von für die Integration verantwortlichen Personen ausgegangen. Alle verantwortlichen Personen werden vom RBAC-System mit Rollen und Rechten ausgestattet. Die Anwendungslogik stellt die Möglichkeiten zur Verfügung, Rechte zu definieren und diese in Anwendungsrollen zusammenzufassen. Diese Anwendungsrollen können dann automatisch, d.h. durch statische Regeln mit der Geschäftslogik verknüpft werden oder sie können Struktureinheiten zur Vergabe zur Verfügung gestellt werden. So bleibt die Kontrolle über die Anwendungen und die Zugriffe darauf, auf Rollenniveau, beim Anwendungsbetreuer. Dieser ist auf der anderen Seite von der Zugriffsverwaltung auf Benutzerebene befreit.

Die bekanntesten RBAC-Modelle, die die Verwaltung von RBAC-Systemen durch das RBAC-Modell selbst beschreiben, gehen von einer Trennung des Modells aus, bei dem administrative Rollen in einer eigenen Hierarchie gespeichert werden. Eine solche Trennung ist nur dann sinnvoll, wenn die Verantwortung für die Teilmodelle geteilt wird. Im Sinne der Benutzungsfreundlichkeit werden in dem hier vorgestellten System die administrativen Rollen

nicht getrennt. Die Rollenverwaltung ist aus Sicht des RBAC-Modells eine Anwendung, auf die entsprechende Rechte vergeben werden.

3.3.2 Sichtenorientierter Modellierungsansatz

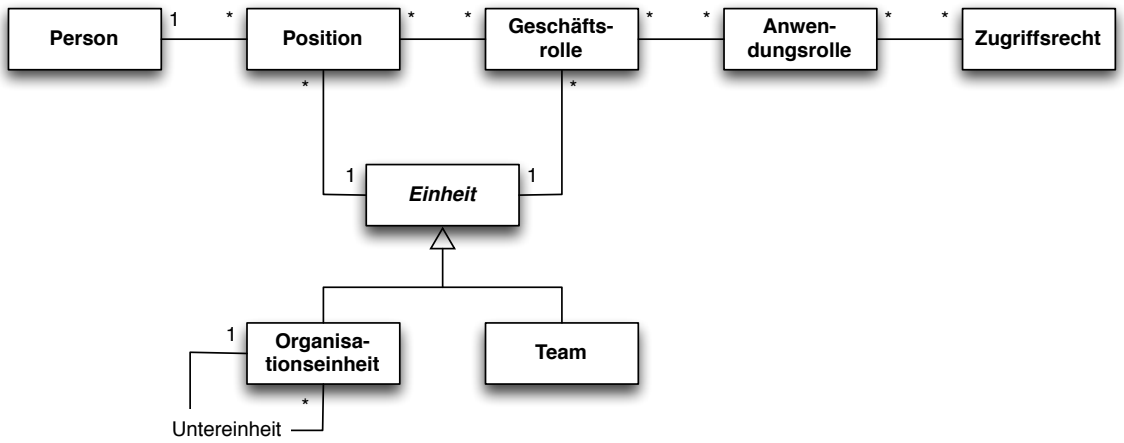

Abbildung 3.3: Klassendiagramm: TUBIS Modell (vereinfacht)

Unterschiedliche Sichten, so genannte ViewPoints, sind im Bereich Software Engineering hinlänglich bekannt [25]. Im Role-Engineering wird sich hingegen in der Regel nur eine einzige Sicht auf das Zugriffsmodell zunutze gemacht. Dabei war das intuitive Verständnis von Rollen eine wesentliche Motivation für die Entwicklung von RBAC überhaupt [23]. Die Bereitstellung unterschiedlicher Sichten auf ein Zugriffsmodell mit der Möglichkeit der Einflussnahme auf die Modellierung eröffnet die Nutzung des Fachwissens völlig unterschiedlicher Akteure innerhalb der Organisation. Die Vorteile eines solchen Ansatzes sind aus der Arbeitswissenschaft bekannt:

- Höhere Akzeptanz des Resultats
- praktikablere Ergebnisse
- Motivation der Mitarbeiter
- bessere Integration in die Arbeitsabläufe

Das RBAC Referenzmodell ermöglicht ohne Erweiterungen oder Änderungen die Realisierung von Sichten auf das Modell. Hierzu ist es üblich, von der Möglichkeit Gebrauch zu machen, Rollenhierarchien zu bilden und diese dann zu typisieren, um dann Sichten auf das Modell durch Auswahl geeigneter Rollentypen erzeugen zu können.

Ein Beispiel für ein solches Rollenmodell ist das so genannte Stanford-Model [24] im Kapitel 4.6 ("Accounting for the Stanford model"). Rollen werden in diesem Modell in "Stanford roles", "Functions", "Tasks" und "Entitlements" untergliedert. Ferner wird in diesem Modell zwischen einer Geschäftssicht (business view) und einer Systemsicht (internal / system view) unterschieden. Dieser Ansatz lässt sich nun erweitern, so wie in dem parallel entwickeltem TUBIS Modell geschehen (Abb. 3.3).

Es kann zwischen folgenden Sichten unterschieden werden:

- Selbstverwaltungssicht
- Identitätssicht
- Organisationssicht
- Anwendungssicht
- Teamsicht
- Organisationsstruktursicht
- Kundendienstsicht
- Rollenadministrationssicht

Alle Sichten können durch ein zentrales Identitäts- und Rollenmodell abgebildet und verwaltet werden. Eine Benutzerin oder ein Benutzer hat in jedem Fall auf die Selbstverwaltungssicht Zugriff und kann ferner abhängig von der jeweiligen Funktion Zugriff auf verschiedene weitere Sichten haben.

Selbstverwaltungssicht

Diese Sicht hat zwei Aufgaben: 1. Transparenz für die Benutzer schaffen; 2. Realisierung von Vertretungen.

Über die Selbstverwaltungssicht sollen dem Benutzer alle im Modell bekannten Daten zur Person zugänglich gemacht werden. Zum einen kann die betreffende Person so die Richtigkeit der Daten überprüfen und entweder selbst korrigieren oder die Korrektur veranlassen. Zum anderen erhält sie so einen Überblick über die im System bekannten Attribute. Idealerweise sollte es möglich sein, eine Aufstellung darüber zu bekommen, welche Attribute von welchen Anwendungen genutzt werden. Somit kann der Benutzer selbst über den Gebrauch des Dienstes entscheiden. Im Sinne der den Datenschutz fördernden Techniken sollte ferner eine Funktion vorhanden sein, die die Konfiguration von pseudonymen Nutzungen von bestimmten Diensten ermöglicht. Über die Selbstverwaltungssicht könnte ich so als Benutzer konfigurieren, dass ich beim Zugriff auf das Berichtesystem keinen Wert darauf lege, mit Namen angesprochen zu werden und somit nur meine Autorisierung ohne Namen übertragen wird oder dass ich grundsätzlich nicht nach privater Nutzung bei der Softwarebestellung gefragt werden möchte, da dies die Bekanntgabe meiner privaten Anschrift nach sich ziehen würde[4].

Im Modell aus Abb. 3.3 ist das eigene Personenobjekt sichtbar sowie die mit diesem Objekt assoziierten Positionen und die mit den Positionen verknüpften Organisations- und Anwendungsrollen. Der Benutzer sieht einen horizontalen Ausschnitt des gesamten Modells mit Ausnahme der an die Anwendungsrollen gebundenen Zugriffsrechte, da diese oft von den Anwendungen intern verwaltet werden und meist auf Grund des Detaillierungsgrades nur einen geringen Informationsgehalt für den Benutzer haben.

[4] An der TU Berlin wurde die Frage nach der Privatanschrift bei der Softwarebestellung so gelöst, dass der Benutzer zwar als TU-Mitglied gegenüber dem Anbieter autorisiert wird, die Liefer- und Rechnungsanschrift jedoch vom Benutzer selbst eingegeben werden muss.

Identitätssicht

Die (Teil-)Identitäten der Organisationsmitglieder werden an verschiedenen datenhaltenden Stellen erfasst und verwaltet. An der Universität betrifft dies die Studierenden sowie die Mitarbeiter. Bei Etablierung eines umfassenden Autorisierungsmanagements dürfen jedoch auch nicht diejenigen Personen vergessen werden, die Zugriff auf Ressourcen der Einrichtung benötigen, die jedoch von keiner der vormals genannten Stellen verwaltet werden. Das können Mitarbeiter von Dienstleistern sein oder an der Universität Nebenhörer. Auch für Gäste der Einrichtung müssen Regelungen getroffen werden. Diese erhalten bei Präsentationen, Beratungen oder Tagungen z.B. Zugriff auf das WLAN.

Die Verwaltung der Identitäten kann am effizientesten mit dafür spezialisierten Anwendungen (z.B. Personalverwaltung) erfolgen. Die Daten müssen dann geeignet mit dem Autorisierungssystem synchronisiert und falls nötig konvertiert werden. Dabei hat es sich bewährt, die Synchronisation für diese Daten nur unidirektional zuzulassen; d.h. Änderungen an Personaldaten werden allein von der Personalstelle gemacht. Zur Vereinfachung der Prozesse kann das Personal Änderungsanträge elektronisch stellen. Die Daten aus den Anträgen können nach Prüfung dann automatisch durch Mitarbeiter der Personalstelle übernommen werden.

Im Modell aus Abb. 3.3 sind die Personen-Objekte und die assoziierten Stellen sichtbar, sofern es sich um Personen aus dem verwalteten Personenkreis (Mitarbeiter, Studierende, Externe) handelt. Für Gäste gilt: Gast-Konten können nur anonymisiert angelegt werden. Die Gast-Konten sind für eine bestimmte Zeit gültig und werden danach automatisch gelöscht. Auch externe Personen werden zeitlich begrenzt angelegt und können bei Bedarf verlängert werden. Die zeitlichen Begrenzungen für Mitarbeiter richten sich nach den Arbeitsverträgen.

Organisationssicht

In der Regel handelt es sich bei der Organisationssicht um die Sicht auf einen kleinen Ausschnitt der Organisation. Die Rollenadministratoren der Organisationseinheiten (auch Strukturverwalter genannt) können Geschäftsrollen anlegen, ändern und löschen, die an ihre Organisationseinheit gebunden sind. Ferner können sie die Mitglieder für die Geschäftsrollen verwalten. Dabei ist es auch möglich, Personen anderer Organisationseinheiten zu Mitgliedern einer Geschäftsrolle der eigenen Einheit zu machen.

Um zu verhindern, dass Organisationseinheiten nicht verwaltet werden können, gilt die Sicherungsregel: Existiert für eine Einheit kein Strukturverwalter, so ist der Strukturverwalter der übergeordneten Einheit für die Untereinheit verantwortlich. In diesem Fall sieht der Strukturverwalter einen größeren Ausschnitt aus dem Modell. Ferner muss jeder Strukturverwalter einen Stellvertreter benennen, um zu garantieren, dass Organisationseinheiten und ggf. Untereinheiten handlungsfähig bleiben.

Im Alltag sind Änderungen im Rollenmodell für die vergleichsweise kleinen Einheiten eher selten. Die Rollenadministratoren müssen vor allem dann reagieren, wenn die Vertretungsregelungen nicht ausreichend sind, weil z.B. Rolleninhaber und Stellvertreter krank sind oder für eine Rolle bislang kein Stellvertreter bestimmt war.

Im Modell aus Abb. 3.3 können Strukturverwalter alle ihrer Organisationseinheit zugeordneten Geschäftsrollen, die damit verknüpften Anwendungsrollen und Mitglieder einsehen. Ferner können sie nach beliebigen Mitgliedern in der Organisation suchen, um diese zu Mitgliedern einer Geschäftsrolle zu machen. Dabei werden ihnen die Attribute angezeigt, die sie benötigen, um die Person eindeutig identifizieren zu können.

Anwendungssicht

Anwendungen besitzen klassischer Weise eine eigene Sicht auf die Organisation, die durch die modellierten Prozesse bestimmt ist. Ein Anwendungsadministrator startet in der Regel in einem leeren Raum. Er ist dann in der Lage, Anwendungsrollen zu modellieren. Diese Anwendungsrollen können dann der Organisation zur Verfügung gestellt werden. Das entsprecht exakt der Sichtweise der Anwendung selbst, die dafür eingerichtet wird, um von Personen aus der Organisation genutzt zu werden.

Im Modell aus Abb. 3.3 sind lediglich die Organisationseinheiten selbst bzw. deren Typisierung sichtbar sowie Standardrollen (dabei handelt es sich um automatisch vergebene Geschäftsrollen) und die eigenen Anwendungsrollen und Zugriffsrechte. Die Sichtweise steht im Kontrast zu der sonst üblichen uneingeschränkten Sicht auf alle Benutzerdaten, die Anwendungsadministratoren sonst besitzen. Die Anwendungsadministratoren schätzen in der Regel jedoch, von der Benutzerverwaltung komplett entlastet zu sein und investieren gerne die Zeit für die Definition des anwendungsbezogenen Rollenmodells. Üblicherweise besitzen die Anwendungen in der TU Berlin weniger als 10 Anwendungsrollen, was den Aufwand der Rollenadministration auch hier relativiert.

Teamsicht

Bei der Teamsicht handelt es sich um eine spezialisierte Organisationssicht. Teams verhalten sich wie Organisationseinheiten. Sie können von Teamverwaltern selbst erstellt werden und dienen der Abbildung von kurzlebigen oder inoffiziellen Einheiten, wie Arbeitsgruppen, Projektteams, etc.

Organisationsstruktursicht

Die Organisationsstruktur, d.h. das Organigramm der Organisation, kann unabhängig vom Rollenmodell gepflegt werden. An der TU Berlin wird das Organigramm aus Daten der Finanzabteilung gewonnen. Die Pflege der Struktur kann wie bei der Identitätssicht unabhängig vom Autorisierungsmodell aufgebaut und dann geeignet importiert werden. Regelmäßig finden Abgleiche der Struktur statt. Auf Änderungen im Strukturmodell muss das Autorisierungssystem reagieren. Wechseln Personen die Organisationseinheit, so verlieren sie automatisch die Geschäftsrollen bei der alten Einheit. Die Rollenzuweisung muss in der neuen Einheit aktualisiert werden. Gleiches gilt bei der Zusammenlegung oder Trennung von Organisationseinheiten. Ein Automatismus für diese Anwendungsfälle wäre wünschenswert, gefährdet jedoch die Konsistenz des Rollenmodells und potentiell auch die Einhaltung der Sicherheitspolitik.

Rollen automatisch aus dem Organigramm abzuleiten, ist riskant und sollte vorab sehr genau durchdacht werden. Wie in [65] beschrieben sind Organigramme in der Regel das Resultat sehr unterschiedlicher Motivationen und spiegeln in den seltensten Fällen eine Hierarchie wider, die 1:1 mit den Geschäftsprozessen in Deckung gebracht werden kann. Aus diesem Grund bietet es sich an, die Struktur aufzubauen und die Rollen auch an dieser Struktur zu verankern. Die Definition der Geschäftsrollen und die Mitgliedschaften in diesen Rollen sollten jedoch davon unabhängig realisiert werden.

Kundendienstsicht

Auch der Support muss im Rahmen des Autorisierungsmanagements bedacht werden. Seine Aufgabe ist in diesem Zusammenhang nicht die Behebung der Hindernisse, d.h. hier die Zuweisung von nicht vorhandenen Rechten, sondern die Unterstützung bei der Analyse, *ob* es sich um ein Rechteproblem handelt und wer wiederum die Berechtigung besitzt, die nötigen Rechte zu vergeben.

Dazu müssen dem Kundendienst sehr weitreichende Rechte in Bezug auf das Autorisierungssystem eingeräumt werden. Nach erfolgter Identifikation des Kunden am Telefon oder vor Ort muss der Kundendienst in die Lage versetzt werden, abzufragen, auf welche Anwendungen ein Kunde Zugriff hat und ggf. für eine fragliche Anwendung, welche Anwendungsrollen dem Benutzer zugeordnet sind. Lesende Zugriffe, beschränkt auf den Kunden, sind für eine Analyse ausreichend. Die Anfragen durch den Kundendienst werden protokolliert, um die Kundendienstmitarbeiter bei Bedarf entlasten zu können und vom Verdacht des Ausspähens von Personen zu befreien. Ferner beschränkt sich die Sicht auf die Rechte der Benutzer. Personenbezogene Daten müssen begrenzt zur Verfügung stehen. Vertragszeiträume müssen beispielsweise nur bei Überschreitung angezeigt werden ("Der Vertrag des Kunden ist abgelaufen. Details müssen mit der Personalabteilung geklärt werden!").

Rollenadministrationssicht

Die Rollenadministrationssicht dient dem 2nd-Level-Support und der Analyse und Planung für das Rollenmodell. Diese Sicht dient der Einrichtung von globalen Rollen, also beispielsweise Geschäftsrollen, die bereits für bestimmte Organisationseinheiten zur Verfügung stehen, um den lokalen Administratoren Arbeit abzunehmen oder Beschlüsse für die Organisation global umzusetzen. Ihre Aufgabe ist auch die Erstellung von Vorlagen, die von den lokalen Rollenadministratoren verwendet werden können.

Die Prinzipien der mehrseitigen Sicherheit fordern für diese Sicht eine Verteilung der Verantwortlichkeit zur Verdachtsentlastung der Rollenadministratoren und zur Minimierung der Angriffsmöglichkeiten gegen das System.

3.3.3 Modellierungswerkzeuge

Eine der wichtigsten Botschaften der IT-Sicherheit ist die Tatsache, dass Sicherheit kein Produkt, sondern ein Prozess ist. Die Zugriffskontrolle ist ein wichtiger Bestandteil dieses Prozesses. Folglich müssen die Modellierungswerkzeuge die Prozesse unterstützen. Produkte oder einmalig durchgeführte Maßnahmen können nur Teil des Prozesses sein.

In vielen Artikeln zum Thema RBAC wird das konkrete Rollenmodell als statisch angesehen. Das RBAC-Modell einer umfassenden Autorisierung ändert sich jedoch stetig, da es z.B. wächst oder sich den neuen Gegebenheiten der Organisation anpasst. Es darf dabei auch nicht vergessen werden, dass sich das Verhältnis der Benutzer zur Nutzung dieses Rollenmodells ändert. Aus diesem Grund ist ein umfassendes Autorisierungsmanagement nicht in Folge eines groß angelegten Modellierungsprojektes mit weitreichenden Schulungen umzusetzen, sondern nach dem Vorbild des Rapid Prototyping oder des eXtreme Programming. Startpunkt ist das kleinste Modell, das funktioniert und das einen Anwendungsfall unterstützt. In dieser Phase ist es zunächst schwer, den eigentlichen Nutzen eines Rollenmodells zu vermitteln. Der Nutzen erschließt sich jedoch mit wachsendem Modell und damit wachsender Komplexität der Rechteverwaltung.

Tabelle 3.1: Sichten auf das TUBIS Modell mit Berechtigungen

	Person	Position	Geschäftsrolle	Anwendungsrolle	Zugriffsrecht	Organisationseinheit
Selbstverwaltung	eigene	eigene	eigene	eigene	—	eigene
Identitätssicht	verwaltete	verwaltete	—	—	—	—
Strukturverwaltungssicht	verwaltete	verwaltete	org.spezif.	•	—	org.spezif.
Anwendungsverwaltung	—	—	—	•	•	•
Teamverwaltung	○	○	•	○	—	org.spezif.
Organisationsstruktur	—	—	—	—	—	•
Kundendienstsicht	○	○	○	○	—	○
Rollenadministration	•	•	•	•	•	•

•: Zugriff auf alle Objekte dieser Klasse
○: Beschränkter Zugriff auf eine definierte Menge von Objekten dieser Klasse
—: Kein Zugriff auf Objekte dieser Klasse

Auch wenn der Prozess mit einem minimalen Rollenmodell startet, sollten die Werkzeuge zumindest so gewählt sein, dass ein späterer Wechsel nicht mehr nötig ist und dass die Infrastruktur mit den Anforderungen an das Autorisierungsmanagement wachsen kann.

Das Modell als Werkzeug

Das verwendete RBAC-Modell stellt ein wesentliches Werkzeug im Prozess dar. Wie schon erwähnt wurde die Rollenmetapher gewählt, um einen intuitiven Zugang für alle beteiligten Parteien zum Thema zu ermöglichen. Damit dieser Vorteil ausgespielt werden kann, muss das konkrete Modell so aufgebaut werden, dass eine gemeinsame Sprache der Beteiligten mit Hilfe des Modells vereinfacht wird. Ziel ist es, über Anwendungsfälle direkt am Beispiel des konkreten Rollenmodells diskutieren zu können. Ein Mittel, um dies zu erreichen ist die bereits vorgestellte Typisierung der Rollen. Legt man bei einer Diskussion ein nicht weiter konkretisiertes NIST RBAC Modell zu Grunde, wird das intuitive Rollenverständnis der Gesprächspartner zu Konflikten führen. Die Systemadministratoren und Anwendungsentwickler werden ein systemnahes Verständnis von Rollen besitzen, wie "Datenbankadministrator für Datenbank X" oder "Tabelle Y Leser" oder "Benutzer mit Zugriff auf Verzeichnis abc/". Die Leiter der Sachbearbeiter und Mitglieder aus den Abteilungen können dieses Rollenverständnis nicht mit ihrem Verständnis zusammenführen für sie sind Rollen z.B. "Kostenstellenverantwortlicher", "Sekretariat der Einrichtung" oder "Projektmitglied". Die Typisierung der Rollen ist ein Werkzeug, das dem gemeinsamen Verständnis dient.

Benutzerschnittstellen

Die Interaktion mit dem Rollenmodell sollte jeder Zeit und auf einfache Weise möglich sein. Hierzu stehen dem Benutzer die in Kapitel 5.1.1 diskutierten Webschnittstellen oder spezialisierte Werkzeuge, wie in Kapitel 3.3.2 beschrieben, zur Verfügung. Jeder Benutzer wird in die Lage versetzt, seinen Teil des Zugriffskontrollmodells zu verwalten. Im einfachsten Fall bedeutet dies: Bestimmung einer Urlaubsvertretung.

eXtreme Role-Engineering Werkzeuge

Das eXtreme Role-Engineering (xRE) Vorgehensmodell löst die Verwaltung der Zugriffskontrolle im Sinne von Modellkonfigurationen ab und definiert Prozesse zur Änderung des Modells. Im Fokus steht dabei als Zielgruppe der "Teilzeit-Rollenadministrator". Die Werkzeuge dienen also nicht vornehmlich einem Expertenteam, das sich hauptberuflich mit Zugriffskontrolle auseinandersetzt, sondern zielen darauf ab, sich das Wissen um Arbeitsabläufe, Personen- und Fachkenntnisse aus dem Anwenderkreis zunutze zu machen, indem diesem Personenkreis eine beschränkte Verantwortung in Bezug auf das Autorisierungsmodell übertragen wird. Dieser Hintergrund muss besonders beim Aspekt der Benutzungsfreundlichkeit der Werkzeuge berücksichtigt werden. So ist diesem Benutzerkreis besonders wichtig, dass sie "nicht aus Versehen etwas kaputt machen" können. Die Werkzeuge leiten deshalb durch einen Prozess, der mit der Frage nach dem "Warum?" beginnt und über Beispiele zum erwarteten Ergebnis ("Was?") zur Realisierung führen ("Wie?"). Am Ende des Prozesses kann das Ergebnis getestet und in einer Simulation ausprobiert werden, bevor die Änderungen in den produktiven Einsatz kommen. Zu jedem Zeitpunkt können Änderungen rückgängig gemacht und Daten angepasst werden. Das Ergebnis ist ein explorationsfreundliches Produkt, das auch ein Einarbeiten nach dem Prinzip Versuch und Irrtum aber am konkreten Beispiel ermöglicht.

Das Vorgehensmodell versucht alle Benutzer mit in den Prozess einzubeziehen. Am Anfang des Prozesses steht die Definition einer Zielbeschreibung (Story). xRE sieht vor, dass jeder Benutzer einen Vorschlag an seinen Rollenverwalter in Form einer Story schicken kann. Die gesammelten Stories werden dann in den xRE Werkzeugen angezeigt und können bearbeitet werden, um schließlich als Ziel für den nächsten Modellierungsprozess genutzt zu werden.

Ein kritischer Punkt bei jeder Implementierungsarbeit ist die Dokumentation. Gerade die Tatsache, dass die Rollenmodellierung im vorgestellten Modell auf relativ kleine Gruppen mit vergleichsweise wenigen Rollen verteilt wird, verführt zu dem Schluss, dass hier eine Dokumentation gar nicht nötig wäre. Insofern sind Protokolldateien oft die einzigen Indizien, die bei Anfragen im Second-Level Support zur Klärung eines Sachverhaltes beitragen können. Die xRE Werkzeuge versuchen, die Dokumentation im Laufe des Prozesses zu erzeugen. Die Story, die am Anfang einer xRE-Sitzung steht, wird ebenso protokolliert wie die definierten Testfälle, die während des Prozesses implementierten Modelländerungen inklusive der Rollennamen und Beschreibungen, die im Prozess von dem Rollenadministrator abgefragt werden. Das Ergebnis ist eine erweiterte Historie, die der Organisationseinheit verfügbar bleibt. Es handelt sich so um menschenlesbare Logbücher, die eine Entwicklung widerspiegeln. Die Dokumentation entsteht so bei der Arbeit im Rollenmodell automatisch.

Die xRE-Werkzeuge werden in Kapitel 4.3 im Detail beschrieben.

3.3.4 Anwendungsintegration

Um eine Anwendung im Rahmen des umfassenden Autorisierungsmanagements zugänglich zu machen, müssen folgende Aufgaben erfüllt werden:

1. Die Anwendung muss im Autorisierungsmodell abgebildet werden.
2. Benutzer- und Rolleninformationen müssen gemäß des Rahmenwerks für die Anwendung zugänglich gemacht werden.
3. Das Portal muss den Zugang zur Anwendung für autorisierte Personen zur Verfügung stellen.
4. Es muss eine Autorisierung von Personen für die Anwendung erfolgen (d.h. es muss eine geeignete Rollenzuweisung stattfinden).

Abbildung im Autorisierungsmodell

Für Anwendungsbetreiber sollte eine eigene Sicht auf das Rollenmodell zur Verfügung gestellt werden. Dieser Ansatz wird im Abschnitt 3.3.2 genauer erörtert. Innerhalb dieser Sicht kann der Anwendungsbetreiber nun Anwendungsrollen definieren, die den Ausschnitt des Zugriffsmodells abbilden, der für die Anwendungsfälle der Anwendung relevant ist.

Datenaustausch zwischen Anwendung und Autorisierungsdienst

Für die Übergabe von Benutzer- und Rolleninformationen stehen zwei architektonische Varianten zur Verfügung, die in der Literatur User-Pull und Server-Pull genannt werden [24, Chapter 1.4.2], [75]. Die Anwendung muss dabei in die Lage versetzt werden, die Autorisierung des Benutzers zu prüfen. Das kann im Fall des User-Pull Verfahrens (Abb. 3.4) dadurch geschehen, indem der Benutzer eine Bestätigung seiner Autorisierung (oft "Credential" oder

"Ticket" genannt) vorab von einem Autorisierungsdienst erhält und diesen Nachweis bei der Nutzung der Anwendung präsentiert. Das Server-Pull Verfahren setzt voraus, dass die Anwendung in die Lage versetzt wird, die Identität des Client zu prüfen. Die Anwendung fragt dann selbst die Autorisierung des Benutzers beim Autorisierungsdienst ab (Abb. 3.5).

Oft werden Mischformen dieser Autorisierungsmodelle implementiert. Single-Sign-On Mechanismen werden in der Regel über ein User-Pull-Verfahren realisiert. Nach erfolgter Authentisierung erhält das Subjekt ein Ticket, das zum Nachweis bereits erfolgter Identifikation präsentiert werden kann. Die Autorisierung kann dann sowohl mittels User-Pull als auch Server-Pull geprüft werden. Dabei können weitere Komponenten, wie der bereits vorgestellte Proxy oder Dekorator zum Einsatz kommen. An Stelle der Anwendung kann hier der Proxy den Server-Pull Mechanismus ausführen und die weitergeleiteten Anfragen an die Anwendung mit diesen Daten dekorieren. Im anderen Fall kann die Anwendung z.B. durch Abfrage eines LDAP-Verzeichnisses oder durch Nutzung von Webservices die Autorisierungsinformationen lesen.

Bei der Integration von Anwendungen werden oft neben einer Anwendungsrolle und einer Benutzerkennung weitere Informationen benötigt. Diese lassen sich grob in zwei Klassen unterteilen:

1. Separation-of-Duty (SoD) Attribute
2. Personenbezogene Attribute

Die SoD Attribute dienen der Präzisierung der Autorisierung. Es kann z.B. relevant sein, ob ein Subjekt selbst Rolleninhaber ist oder die Rolle in Vertretung ausfüllt. Der Zugriff auf eine Ressource kann ferner von der Zugehörigkeit zu einer Organisationseinheit abhängig sein oder davon, ob das Subjekt weitere Rollen besitzt, die sich beispielsweise ausschließen.

Die personenbezogenen Attribute können von der Anwendung gefordert sein, um die Qualität des Benutzerdialogs zu verbessern (z.B. Ansprechen des Benutzers mit Namen) oder weil sie im Arbeitsprozess aus organisatorischen oder rechtlichen Gründen notwendig sind. Beim Entwurf der Datenübergabe sollte auch im Hinblick auf die Frage nach der Datensparsamkeit klar unterschieden werden, welche Gründe für die Datenübergabe vorliegen. So legen Benutzer zugunsten eines pseudonymen Zugangs evtl. keinen Wert darauf, mit bürgerlichen Namen angesprochen zu werden. In Hinblick auf einen späteren Nachweis reicht in der Regel auch die Speicherung der Benutzerkennung aus. Sie kann im Bedarfsfall wieder einer natürlichen Person zugeordnet werden.

Zugang über das Portal

Sind die Voraussetzungen für die Anwendung erfüllt und ist diese im Rahmenwerk integriert, muss die Anwendung im Portal sichtbar gemacht werden. Dieser Schritt ist entscheidend für den Benutzer, weil hier ein wesentlicher Mehrwert für ihn entsteht. Es existiert eine zentrale Stelle, über die er auf alle wesentlichen für seine Aufgaben nötigen Anwendungen zugreifen kann. Wird eine neue Anwendung zur Verfügung gestellt, so erfährt der Benutzer dies über einen zentralen Punkt.

Vor diesem Hintergrund sollte das Portal für die Benutzer so angelegt sein, dass sie über ihren Zugangspunkt in die Lage versetzt werden, weitergehende Informationen zu den verfügbaren Anwendungen zu erlangen. Hierzu gehört z.B. auch, einen Überblick darüber zu haben, welche Anwendungen zur Zeit nicht genutzt werden können, weil sie aus technischen oder

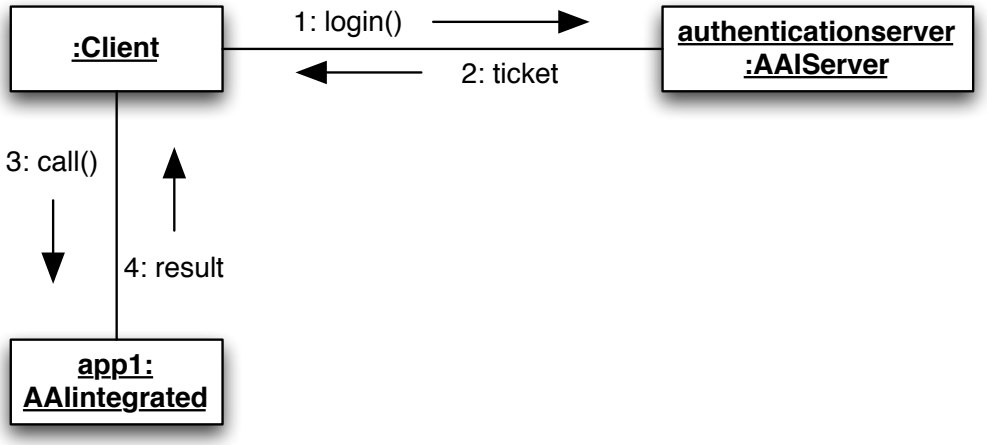

Abbildung 3.4: Kollaborationsdiagramm: User-Pull Autorisierung

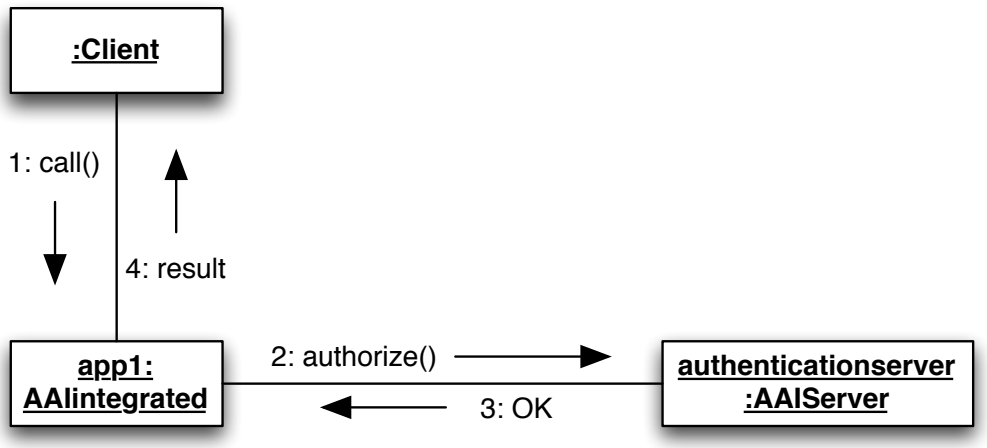

Abbildung 3.5: Kollaborationsdiagramm: Server-Pull Autorisierung

organisatorischen Gründen nicht zur Verfügung stehen oder weil man zu deren Nutzung nicht autorisiert ist. Dies ermöglicht eine Rückkopplung mit den lokalen Rollenverwaltern ("Ich könnte meine Aufgaben effizienter bewältigen, wenn ich Zugriff auf diese Anwendung hätte.").

3.3.5 Workflowintegration

Bezüglich der Workflowintegration, also dem Einbeziehen in Arbeitsprozesse sind zwei Arten von Prozessen zu unterscheiden:

1. Prozesse des Identitäts- und Autorisierungsmanagements selbst.
2. Prozesse, die durch die AA-Infrastruktur unterstützt werden, d.h. die durch die AAI verbessert oder ermöglicht werden.

Zu den Prozessen des Identitäts- und Autorisierungsmanagements gehören:

- Provisioning
- De-Provisioning
- Modellpflege
- Authentisierung und Autorisierung

Diese Prozesse sind in den Kapiteln 3.3.2 und 5.2.1 beschrieben. Die zum xRE gehörenden Arbeitsprozesse sind in Kapitel 4 ausgeführt. Dem gegenüber stehen die Prozesse, die selbst Nutzer des Autorisierungsmanagements sind. Diese Prozesse sollen durch die Integration mit dem Autorisierungsmanagement verbessert werden. Dabei kann eine Verbesserung die Erhöhung von Effizienz, die Verbesserung von Benutzungsfreundlichkeit oder die Qualitätssteigerung bedeuten. Einige Prozesse können auf der anderen Seite erst durch Einführung einer AAI digitalisiert werden, weil hierdurch erst die rechtlichen, organisatorischen oder technischen Voraussetzungen geschaffen werden.

Für die Implementierung von Workflows stehen spezialisierte Werkzeuge (Workflow Management Systems, WfMS) zur Verfügung. [24] widmet dem Thema "RBAC for WfMSs" ein komplettes Unterkapitel. In einer Diplomarbeit zum Thema "Modellierung eines Workflow Management Systems innerhalb des rollenbasierten Autorisierungssystems TUBIS" [56] wurde vor allem deutlich, dass für den produktiven Einsatz von WfMSs zunächst erhebliche Investitionen in Form von Schulung/Exploration, Aufbau der technischen Infrastruktur und Implementierung zu tätigen sind. Eine Standardisierung durch den Einsatz eines WfMS ist zwar erstrebenswert und Möglichkeiten, wie die grafische Bearbeitung von Prozessabläufen im Programm wären wünschenswert, doch kommen solche Vorteile erst zum Tragen, wenn ein bedeutender Teil der Prozesse über das WfMS abgebildet sind.

Demgegenüber steht eine sehr einfache Abbildung von Arbeitsprozessen mit Hilfe von kleinen und einfachen Webanwendungen, die in verschiedenen Programmiersprachen, basierend auf unterschiedlichen Bibliotheken und sogar auf verschiedenen Plattformen implementiert werden können.

Ausstattung mit Zugriffsrechten

Die Ausstattung der Benutzer mit den erforderlichen Zugriffsrechten ist *das* zentrale Thema des Autorisierungsmanagements. An dieser Stelle soll erwähnt werden, dass die Ausstattung mit Zugriffsrechten ein wesentlicher Bestandteil der Integration der Anwendung ist. Die zu treffende Entscheidung ist nun, *wie* die Berechtigungen an Rollen gebunden werden und welche Personen Mitglieder in welchen Rollen werden sollten.

Der klassische Ansatz wurde im Kapitel 2.3 vorgestellt. Einen von mir entwickelten alternativen Ansatz werde ich in Kapitel 4 im Detail vorstellen und im Abschnitt 4.7.2 mit dem Role-Mining, einem ähnlichen Ansatz vergleichen, der jedoch bei genauerer Betrachtung auf gänzlich anderen Grundlagen basiert und sich somit im Vorgehen stark unterscheidet.

3.4 RBAC-Muster

Das Ziel von Mustern (Software-Pattern) ist die Systematisierung von Fachwissen zur Schaffung einer gemeinsamen Sprache, das zur Verfügungstellen einer Liste von möglichen Lösungsansätzen und die Nutzbarmachung von Expertenwissen für weniger erfahrene Entwickler. Der Softwareentwurf auf der Architekturebene entleiht die Idee aus der klassischen Architektur. Bestimmte Elemente, wie Türen oder Fenster werden beim Architekturentwurf nicht jeweils von Grund auf neu spezifiziert. Vielmehr ist das "Muster Tür" allgemein bekannt und wird im jeweiligen Kontext nur verfeinert bzw. spezialisiert. Nach diesem Vorbild wurden Muster für gängige Konstruktionen in der Softwaretechnik identifiziert und beschrieben, wie z.B. Proxy, Adapter, Interpreter usw. Das Aufbereiten von RBAC-Wissen in Form von Pattern hat sich bislang (noch) nicht durchgesetzt, obwohl die Idee von Mustern auch auf das Umfeld RBAC übertragbar ist.

Im Folgenden wird motiviert, dass Muster auf RBAC-Systeme auf unterschiedlichen Abstraktionsniveaus angewendet werden können. Der Diskurs wird geschlossen mit Informationen zur Anwendung von Mustern im aktuellen TUBIS-System und der Fragestellung, in wieweit RBAC-Muster mit dem xRE-Ansatz kollidieren.

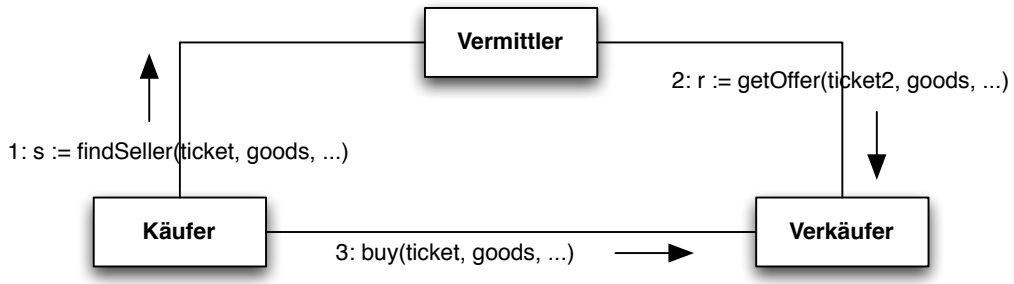

Abbildung 3.6: Objektinteraktionsgraph: Vermittler-Muster nach [31]

Muster in der Softwareentwicklung haben meist ein sehr ähnliches Abstraktionsniveau. Sie dienen dem Erstellen des Klassenmodells. Aus den Klassen können dann Klassen im Softwaresystem abgeleitet werden. In diesem Sinne wurden Muster in den Kapiteln 5.1 und 5.2 eingesetzt. In [31] entwarf Thomas Gebhardt mit mir gemeinsam RBAC-Muster für ein ganz spezielles Anwendungsgebiet (B2B), basierend auf einem von uns vorgegebenen Rahmenwerk

(Abb. 3.6). Dabei machten wir massiv vom Proxy-Muster Gebrauch und übersetzten bekannte Muster, wie z.B. das Broker-Muster in unser Rahmenwerk. Das vorgestellte Muster "Vermittler" entspricht von seiner Funktion her beispielsweise wieder einem Proxy-Muster.

Innerhalb des konkreten RBAC-Modells finden verschiedene bekannte Muster Anwendung, wie das Prototyp-Muster für die Bildung von Vorlagen, das Kompositum für die Strukturbildung von Rollen- oder Organisationshierarchien, das Interpreter-Muster z.B. bei der Definition von Nebenbedingungen.

Betrachtet man das Rollensystem und seine Umgebung, so kann die ADF selbst als Vermittler verstanden werden, der die Zugriffsobjekte des Modells von den AEF der Programme entkoppelt. Wie ein Dekorierer bei der Umsetzung des User-Pull Modells zum Einsatz kommt, ist im Detail in Kapitel 5.1.6 gezeigt.

Es ist jedoch auch möglich, Muster zu bilden, die allgemein auf RBAC-Modellen gelten. Viele Fachartikel aus diesem Umfeld könnten in einer solchen Mustersammlung zusammengestellt werden:

Backup-Role: Die Backup-Role wird z.B. in [65] angeführt und wird in ähnlicher Form auch in TUBIS benutzt.

Rollendelegation: In [6] wird ein Rahmenwerk zur Rollendelegation vorgestellt. Delegierbare Rollen könnten als Muster definiert werden.

Administration von RBAC: Auch die in 2.2.5 angeführten unterschiedlichen Modelle zur rollenbasierten Administration von RBAC-Modellen könnten als Muster zur Verfügung gestellt werden.

Dies sind nur einige Beispiele, die verdeutlichen sollen, dass Muster für RBAC-Modelle bereits existieren, nur noch nicht in dieser Form zusammengestellt wurden. Üblich ist hingegen, in diesem Bereich über Modellarten zu sprechen. Diese vereinigen aber meist mehrere Aspekte, die einzeln betrachtet Muster darstellen.

3.4.1 Muster und eXtreme Role-Engineering

Im eXtreme Programming werden Software-Muster explizit als Mittel zur Diskussion im Entwicklerteam empfohlen. In diesem Bereich nutzt das Vorgehensmodell die Muster. Dieses Modell lässt sich leider nicht direkt auf das xRE übersetzen. Allerdings können Muster bereits auf die Umgebung angewandt sein, in der ein xRE Prozess abläuft. Auch ist es möglich, Muster-"Wissen" in die xRE Werkzeuge zu integrieren, um so spezialisierte Rollenkonstruktionen zu modellieren. Die einfachste Art von Mustern, nämlich die Rollenvorlagen, werden explizit von xRE unterstützt. Die Vorlagen werden halbautomatisch durch Vergleich von existierenden Vorlagen und aktuellen Modellierungen erweitert.

3.5 Anwendung mehrseitiger Sicherheit

Wie wichtig die Umsetzung von Prinzipien der mehrseitigen Sicherheit sind, zeigen die Datenschutzskandale der letzten Zeit [51]. Diese Vorfälle zeigen aber auch, wie schwierig es ist, Mittel für die Umsetzung von "besserem Datenschutz" zu mobilisieren. Denn anders als erwartet, ist Datenschutz kein Verkaufsargument. Die logische Folge für die Verletzung der

Persönlichkeitsrechte tausender Kunden müsste eine massenhafte Abwanderung zu Konkurrenzunternehmen sein. Diese Reaktion bleibt jedoch aus. Auch mit personellen Konsequenzen tut man sich schwer. Es bleibt der rechtliche Druck auf die Organisationen. Der ist jedoch schwach. Das zeigt, dass vor allem Aufklärungsarbeit gefordert ist. Das Interesse einer Organisation an Vertraulichkeit ihrer Daten (und dazu gehören unter anderem personenbezogene Daten), muss herausgestellt werden und jeder einzelne Bürger muss viel mehr als bisher in Bezug auf seine personenbezogenen Daten sensibilisiert werden. Eine schwierige Aufgabe, während im Zuge der vermeintlichen Terrorbekämpfung die Persönlichkeitsrechte der Bürger mehr und mehr eingeschränkt werden.

Vor diesen Herausforderungen kann der nun folgende Abschnitt nur eines tun: Er kann zeigen, was technisch / organisatorisch machbar ist und damit verhindern, dass Diskussionen um den Schutz von Informationen mit einem: "Wir würden ja gerne, aber was soll man denn machen?", enden.

An dieser Stelle möchte ich nicht versäumen, auf die Arbeiten von Qingfeng He hinzuweisen [40, 39], die sich mit dem Datenschutz in RBAC-Modellen auseinandersetzen.

Die Modellierung eines fehlerfreien Zugriffsmodells muss durch den Schutz des Modells und der im Autorisierungssystem zusammenwirkenden Komponenten ergänzt werden. Ein zentrales Autorisierungssystem stellt ein zentrales Angriffsziel dar. Die Annahme, die Bedrohung in IT-Systemen würde jeweils nur von einer Seite ausgehen, ist falsch. Der Schutz der Ressourcen vor unerlaubten Zugriffen von Benutzern allein ist nicht ausreichend. Es gilt nicht nur die Interessen der Anbieter zu schützen, sondern wie die stetig steigende Zahl von Fishing-Attacken zeigt, auch den Nutzer vor betrügerischen (vermeintlichen) Anbietern. Eine nicht zu unterschätzende Gefahr geht ferner von Mitarbeitern innerhalb einer Organisation aus, wobei hier noch zwischen vorsätzlichen und fahrlässigen Schädigungen unterschieden werden muss. Jedes eingesetzte Softwareprodukt stellt potentiell eine Gefahr dar. Programme können gewollt oder ungewollt Schadfunktionen enthalten oder schlicht fehlerhaft sein und somit direkt Schaden verursachen bzw. Angreifern Möglichkeiten für Attacken liefern.

Im Rahmen der wissenschaftlichen Arbeit am umfassenden Autorisierungsmanagement wurde betrachtet, wie die Authentisierung, Autorisierung und das Auditing rollenbasiert und pseudonym implementiert werden können. Abhängig von den Geschäftsprozessen und den jeweiligen Anwendungen eröffnet RBAC die Möglichkeit, einen Nachteil eines zentralen Portalzugangs zu minimieren. Werden alle Anwendungen von einem zentralen Portal aus angesteuert und über ein zentrales Autorisierungsmanagement gelenkt, so fallen dabei zwangsweise in den zentralen Systemen Daten an, die die Erstellung kompletter Bewegungsprofile ermöglichen würden. Die pragmatische Lösung des Problems wäre eine Vereinbarung mit dem Betreiber, die die Pseudonymisierung der Protokolldaten verlangt. Reicht eine solche Vereinbarung nicht aus, ist es möglich, durch Verteilung der Informationen beim Authentisierungsvorgang mit einer Smartcard eine sichere aber pseudonymisierte Identifikation zu erreichen.

3.5.1 Pseudonyme Authentisierung mittels Smartcard

Im Rahmen des Projektes Campuskarte an der TU Berlin wurde das pseudonyme Authentisierungsverfahren entwickelt, das ich in [44] vorstellte.

Abb. 3.7 zeigt die am Verfahren beteiligten Maschinen und die Verteilung der Komponenten. Das Protokoll ist im Kollaborationsdiagramm (Abb. 3.8) dargestellt.

Das Protokoll kann gemäß Abb. 3.8 wie folgt beschrieben werden. Voraussetzung ist eine SSL-gesicherte Verbindung zwischen Webbrowser und Webproxy:

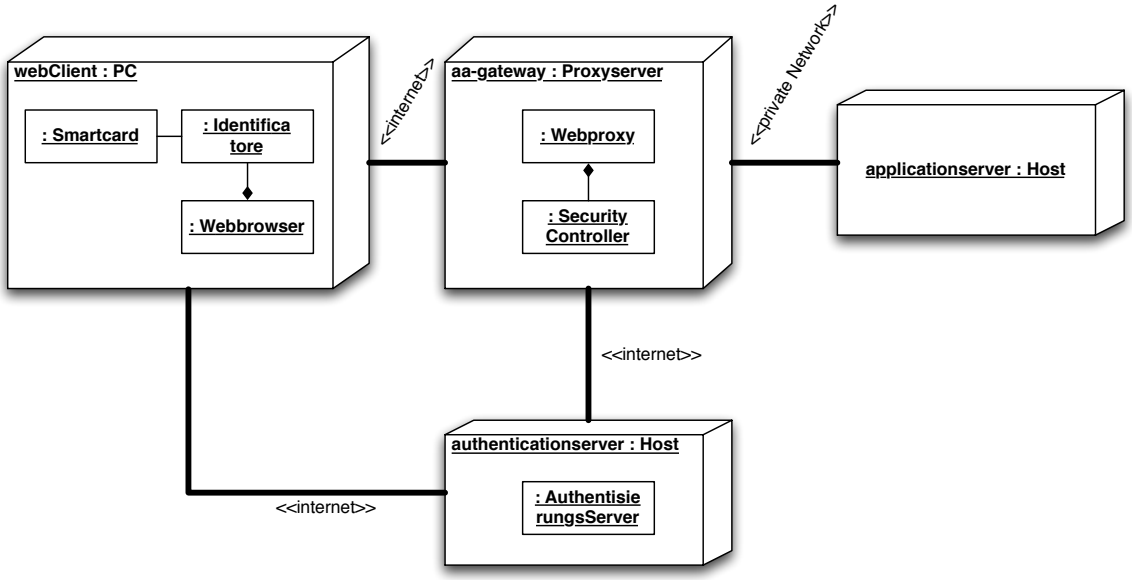

Abbildung 3.7: Verteilungsdiagramm: Pseudonyme Authentisierung

1: get(page): Der Benutzer ruft über seinen Webbrowser eine geschützte Seite auf. Die Anfragen werden über einen Webproxy geleitet.

2: get(page): Die Anfrage des Browsers wird an die Komponente "SecurityController" weitergeleitet, die im Kontext des Webproxy läuft.

3: skSEC, authpage, nrv, CAuthServer: Der SecurityController antwortet mit einem 4-Tupel:

skSEC: Sessionkey des Security Controller

authpage: Authentisierungsseite mit dem zu startenden Servlet "Identificatore"

nrv: eine nicht wiederkehrende zufällige Zahl

CAuthServer: Zertifikat des Authentisierungsservers.

4: skSEC, authpage, nrv, CAuthServer: Die Webseite mit den beschriebenen Attributen wird an den Webbrowser übermittelt.

5: startApplet(skSEC, nrv, CAuthServer): Das Servlet "Identifiactore" wird gestartet. Es erhält das Tripel `(skSEC, nrv, CAuthServer)` als Parameter.

6: reqRnd(skSEC, CAuthServer, nrv): Die Methode `reqRnd()` wird auf der Javacard gestartet. Diese Methode arbeitet wie folgt:

1. Prüfung des `CAuthServer` Zertifikats. Die Attribute des Zertifikats bestätigen die Berechtigung zur Durchführung der Methode.
2. Die Zufallszahl `rndSC` der Smartcard wird generiert.
3. Ein zweiter Sitzungsschlüssel `skSC` wird generiert.

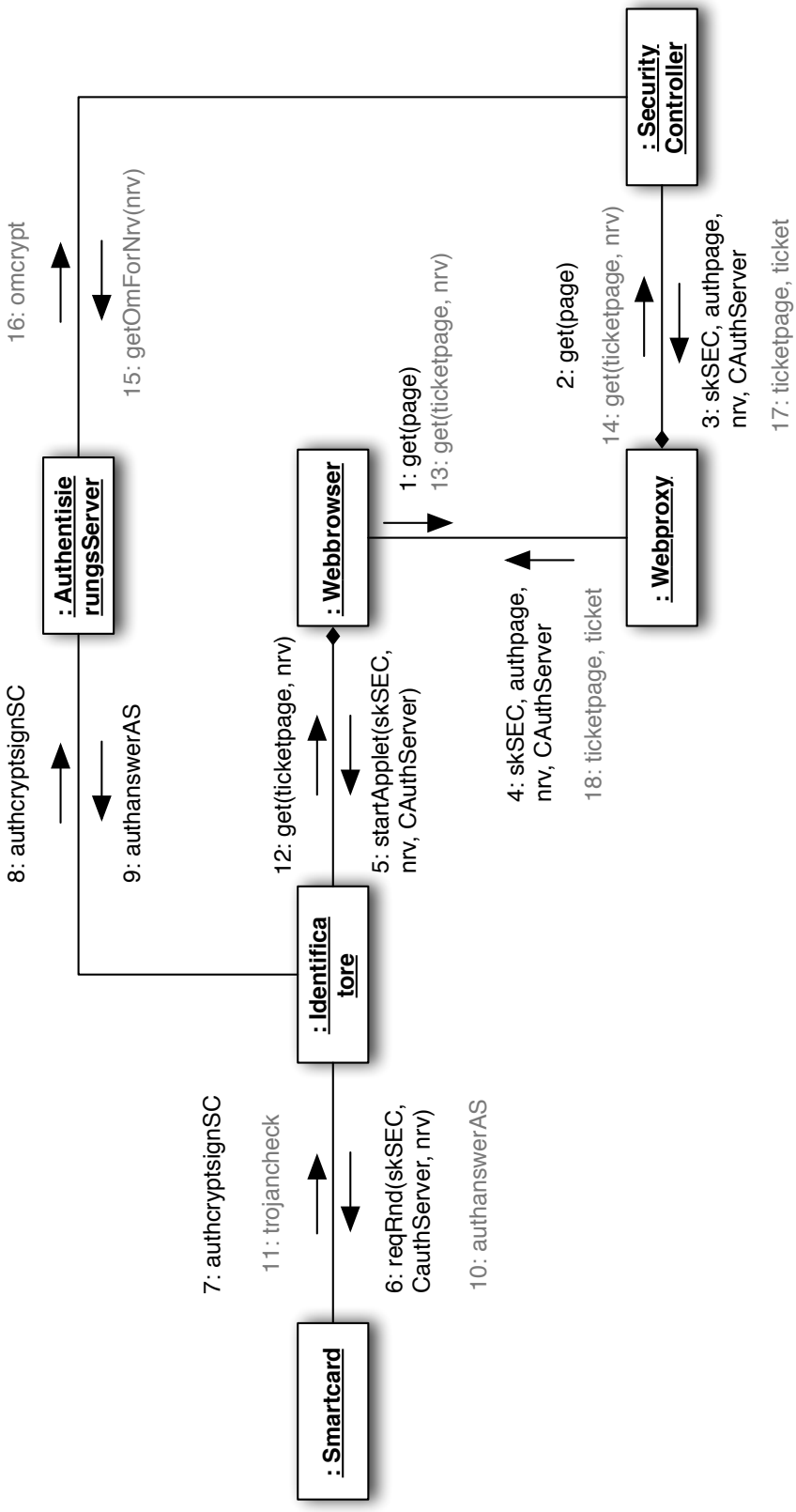

Abbildung 3.8: Kollaborationsdiagramm: Pseudonyme Authentisierung

4. Das Ordnungsmerkmal (Eindeutige Benutzerkennung) `OM` wird mit `skSEC` verschlüsselt und in `omcrypt` gespeichert.
5. Das Tupel `(omcrypt, rndSC, skSC, nrv, CID)` wird gebildet, wobei `CID` die Identifikationsnummer der Smartcard ist, über die die Gültigkeit der Karte ermittelt werden kann.
6. Das Tupel wird mit dem öffentlichen Schlüssel aus `CAuthServer` verschlüsselt und von der Smartcard signiert. Das Ergebnis wird in `authcryptsignSC` abgelegt.

7: authcryptsignSC: Das Ergebnis der Methode wird von der Smartcard an das Applet übergeben.

8: authcryptsignSC: Das Servlet leitet das Ergebnis an den Authentisierungsserver weiter. Dieser führt die folgende Methode aus:

1. Entschlüsselung von `authcryptsignSC`.
2. Abfrage des Kartenstatus mit Hilfe der extrahierten `CID`.
3. Anforderung des Zertifikats `CSc` der Karte mittels `CID`.
4. Prüfen der Gültigkeit von `CSc`.
5. Prüfung der Signatur über `authcryptsignSC` gegen `CSc`.
6. Speicherung des Tupels `(nrv, omcrypt, rndSC, skSC)` für einige Minuten.
7. Verschlüsselung von `rndSC` mit `skSC`: `authanswerAS = rndSC.crypt(skSC)`

9: authanswerAS: Der verschlüsselte Zufallswert wird an das Applet gesendet.

10: authanswerAS: Das Resultat wird an die Smartcard übergeben, die nun vergleichen kann, ob nach Entschlüsselung von der Antwort mit `skSC` wieder `rndSC` entsteht. Damit ist nachgewiesen, dass es sich bei der Gegenstelle um den Besitzer des privaten Schlüssels von `CAuthServer` handelt.

11: trojancheck: Zur Verifizierung des Ergebnisses sendet die Smartcard Teile eines zuvor durch den Benutzer gespeicherten Kennworts. Der Benutzer kann so erkennen, dass die Infrastruktur, mit der er arbeitet, von seiner Smartcard verifiziert wurde.

12: get(ticketpage, nrv): Das Applet fordert die eine Seite mit einem Ticket an. Zur Identifizierung wird `nrv` übergeben. Dann wird das Applet beendet.

13: get(ticketpage, nrv): Die Anfrage wird vom Browser an den Webproxy weitergeleitet.

14: get(ticketpage, nrv): Der Webproxy stellt die Anfrage an den Securitycontroller.

15: getOmForNrv(nrv): Der Securitycontroller fragt das `OM` mittels des übergebenen Wertes `nrv` ab. Mit Hilfe von `nrv` kann das zuvor gespeicherte Tupel gefunden und `omcrypt` extrahiert werden.

16: omcrypt: Wird an den Securitycontroller zurück gesendet. Der Securitycontroller kann das `OM` durch das Entschlüsseln von `omcrypt` mit `skSEC` ermitteln.

17: ticketpage, ticket: Eine Ergebnisseite mit Ticket in Form eines verschlüsselten, signierten Cookie wird zurück gesendet.

18: ticketpage, ticket: Das Ergebnis wird zum Browser weitergeleitet. Die Authentisierung ist abgeschlossen. Es kann eine Weiterleitung auf die ursprünglich angeforderte Seite page erfolgen.

Am Ende der Authentisierung sind die Informationen über den Vorgang auf verschiedene Parteien verteilt. Der Authentisierungsserver kennt die CID, mit der er die Kartengültigkeit prüfen und das Zertifikat mit dem Public-Key der Karte ermitteln konnte. Das OM liegt hier jedoch nur verschlüsselt vor. Die Entschlüsselung gelingt nur dem Securitycontroller, der damit das Ticket ausstellen kann. Der Identificatore vermittelt zwischen Smartcard und Umgebung, leitet jedoch die personenbezogenen Daten nur verschlüsselt weiter. Die Smartcard selbst ist der Container für die Daten. Durch die Verwendung von Attributzertifikaten wird sichergestellt, dass die Daten nur an zertifizierte Stellen weitergegeben werden. Auf der anderen Seite lässt sich die Echtheit der Karte durch Prüfung mit dem Public-Key aus dem Smartcard-Zertifikat prüfen.

3.5.2 Mehrseitig sicherer Zugriff auf Smartcard-Daten

Der typische Einsatz von Smartcards besteht in der Identifizierug des Benutzers vor einem Hintergrundsystem. Nach einer erfolgreichen Authentisierung kann der Betreiber des Hintergrundsystems davon ausgehen, dass der Benutzer im Besitz der Chipkarte ist und zudem den PIN oder das Passwort kennt, mit dem die Karte aktiviert werden kann. Der Zugriff des Hintergrundsystems auf die Daten auf der Chipkarte ist in der Regel über einen symmetrischen Schlüssel gesichert. Ist also das System im Besitz des Schlüssels, darf es auf die Daten auf der Chipkarte zugreifen. Heute verwendete Chipkarten bieten in der Regel auch die Möglichkeit mit verschiedenen Schlüsseln zu arbeiten.

Beim Entwurf der TU Campuskarte in der ersten Version wurde dieses Prinzip erweitert. Mit einem Cardlet auf einer Javacard besteht die Möglichkeit, eine beidseitige Authentisierung mit asymmetrischen Schlüsseln durchzuführen (im Abschnitt 3.5.1 dargestellt). Die Zugriffsrechte, die das Hintergrundsystem gegenüber der Smartcard besitzt können dann über Attribute im Zertifikat geprüft werden [94]. Der Betreiber eines Hintergrundsystems wird vom Betreiber der Infrastruktur zertifiziert. In Abstimmung z.B. mit dem behördlichen Datenschutz kann definiert werden, welcher Betreiber welche Zugriffe auf Daten der Smartcard erhält. Entsprechend der Rechte werden Attribute in einem X.509v3 Zertifikat gesetzt, das das Hintergrundsystem zur Authentisierung gegenüber der Karte übermitteln muss. Die Karte kann dann über ein Challange-Response-Verfahren prüfen, ob das Hintergrundsystem auch im Besitz des privaten Schlüssels ist, der zum vorgezeigten Zertifikat gehört. Die Smartcard muss lediglich das CA-Zertifikat des Infrastrukturbetreibers kennen.

Dieses Verfahren hat verschiedene Vorteile:

- Personenbezogene Daten können auf der Smartcard gespeichert bleiben. Der Benutzter hat physikalische Kontrolle über den Zugriff auf die Daten. Solange er die Smartcard mit sich führt, kann kein System auf die Daten zugreifen.

- Rechte für Betreiber können ohne Änderungen an der Logik auf der Karte sehr dynamisch angepasst werden. Damit ist es möglich, sehr große Infrastrukturen zu betreiben und auf wechselnde Anforderungen zu reagieren.

Jedoch gibt es auch Grenzen beim Einsatz solcher Attributzertifikate auf Smartcards:

- Smartcards können keine Rückruflisten von Zertifikaten verwalten. Das Verarbeiten einer Rückrufliste auf einer handelsüblichen Javacard kann zur Zeit nicht in akzeptabler Geschwindigkeit durchgeführt werden. Stattdessen müssten alternative Protokolle verwendet werden, die lediglich auf die Gültigkeit eines Zertifikats angewiesen sind.
- Da die Samrtcards selbst keine Benutzerschnittstelle besitzen, ist für den Benutzer nicht transparent, welche Zugriffe auf die Karte tatsächlich stattfinden.
- Leistungsstarke Javacards sind heute noch erheblich teuerer als vergleichbare Smartcards, die mit Standardalgorithmen ausgestattet sind.

3.5.3 Pseudonyme Autorisierung

Ein zentrales Zugriffsverwaltungssystem ist jeder Zeit in der Lage, vollständige Bewegungsprofile von Benutzern aufzuzeichnen, da zu jeder Anwendungsnutzung Benutzer- und Anwendungsrollendaten abgefragt werden (Subjekt, Operation, Objekt). In [47] stellte ich drei mögliche Implementierungen eines AEF-ADF Rahmenwerkes zusammen, das die Zugriffsanfragen verschleiert und verteilt. Aus den drei Lösungsansätzen präsentiere ich im Folgenden ein auf Chaume Mixen [13] basierendes Konzept im Sinne der "klassischen" mehrseitigen Sicherheits-Lösung.

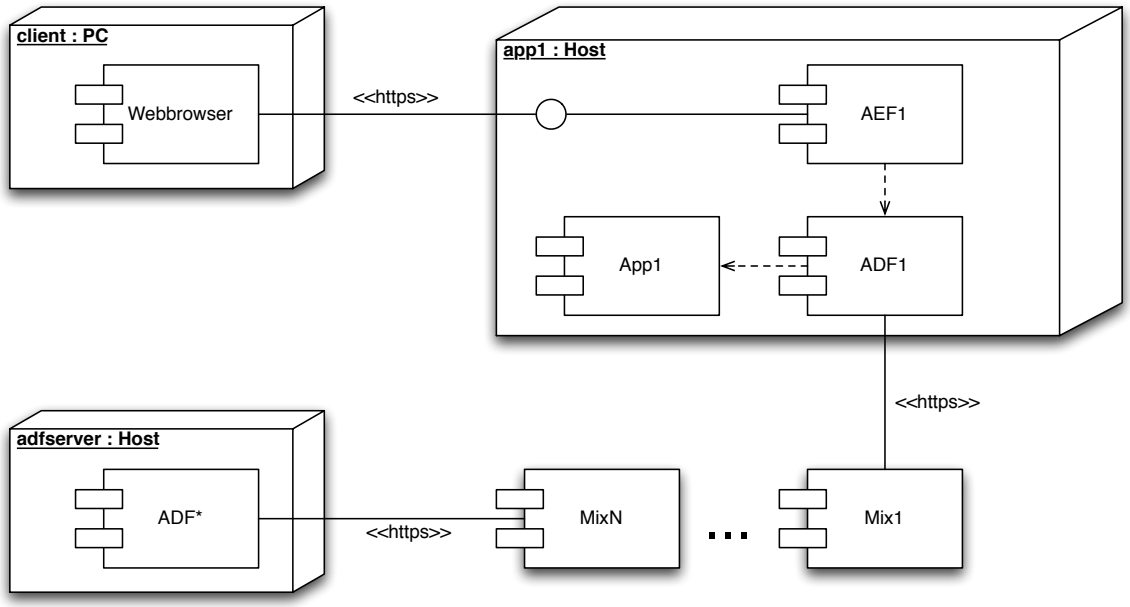

Abbildung 3.9: Verteilungsdiagramm: Pseudonyme Autorisierung ("Hiding of Structure-Application-Mapping" [47])

Abb. 3.9 zeigt die Verteilung der Komponenten. Der Ansatz geht davon aus, dass jede Anwendung einen Teil des Zugriffskontrollmodells in einer lokalen ADF_n hält. Die Geschäftsrollenlogik sowie die Mitgliedschaften der Benutzer in den Geschäftsrollen befinden sich in

der ADF^*. Die ADF_n-Instanzen können Anfragen bezüglich der Benutzer und der Geschäftsrollen über die Mixe-Kaskade stellen. Durch die zufällige Vermischung der Anfragen in den Mixen kann auf Seiten der ADF^* nicht mehr rekonstruiert werden, auf welche Anwendungen ein Benutzer zugegriffen hat und welche Operationen er dort ausgeführt hat.

Beispiel für einen Protokollablauf:

1. AEF_1 sendet eine Autorisierungsanfrage an ADF_1.
2. ADF_1 stellt eine Liste von möglichen Geschäftsrollen zusammen, die Zugriff auf die angeforderte Operation hätten. Für eine effektivere Verschleierung der Benutzerbewegungen sollte die Liste mit zufälligen weiteren Rollen angereichert werden.
3. Die Rollenliste wird zusammen mit dem zu prüfenden Benutzer durch die Mixen an die ADF^* gesendet.
4. ADF^* kann die Mitgliedschaft des Benutzers zu den Rollen prüfen.
5. Es wird eine Liste der Rollen durch die Mixen zurück geschickt, in denen der Benutzer Mitglied ist.

Weitere Methoden

Die weiteren in [47] veröffentlichten Methoden beruhen auf einem Blinding-Mechanismus und der Verteilung auf verschiedene ADF-Instanzen mit gemeinsamer Datenbasis und auf einer Broker-Architektur, die die Anfragen auf verschiedene ADF-Instanzen verteilt. Auch hier greifen die ADF-Entitäten auf eine gemeinsame Datenbasis zu. Bei diesen Verfahren verlagert sich das Problem auf die mögliche Erstellung von Nutzerprofilen durch die Datenbankinstanz. Dies ließe sich lösen, indem die Datenstruktur so gewählt wird, dass durch die Anfragen keine Informationen durch den Datenbankbetreiber gewonnen werden können oder indem, wie ursprünglich geplant, die Daten im Dateisystem liegen und mit Mitteln des Betriebssystems eine Protokollierung der Zugriffe verhindert werden kann.

3.5.4 Pseudonymes Auditing mit Integrationsschutz

In seiner Diplomarbeit [34] entwirft Klaus Hamann einen sicheren Logging-Mechanismus und implementiert diesen prototypisch. Er nimmt dabei Bezug darauf, wie ein "rollenbasiertes Logging" realisiert werden könnte. Die Arbeit wurde mit Unterstützung durch Klaus Nagel von mir betreut.

Die Arbeit geht davon aus, dass die Anwendungen eine Programmbibliothek für das Logging verwenden. Die Aufgaben dieser Bibliothek können auch ausgelagert werden. So könnte es z.B. einen Logging-Dienst geben, der auf Basis von secure Webservices arbeitet. Der Prototyp hat jedoch den Vorteil, dass die Verschlüsselung der Logdaten bereits in der Applikation erzeugt wird und so die Klartextinformationen die Anwendung nicht verlassen.

Die zu protokollierenden Nutzdaten werden signiert und verschlüsselt. Dabei wird die Signatur (der mit dem privaten Schlüssel verschlüsselte Hashwert) und der mit einem öffentlichen Schlüssel verschlüsselte Sessionkey jeweils getrennt von den mit dem Sessionkey verschlüsselten Nutzdaten aufbewahrt.

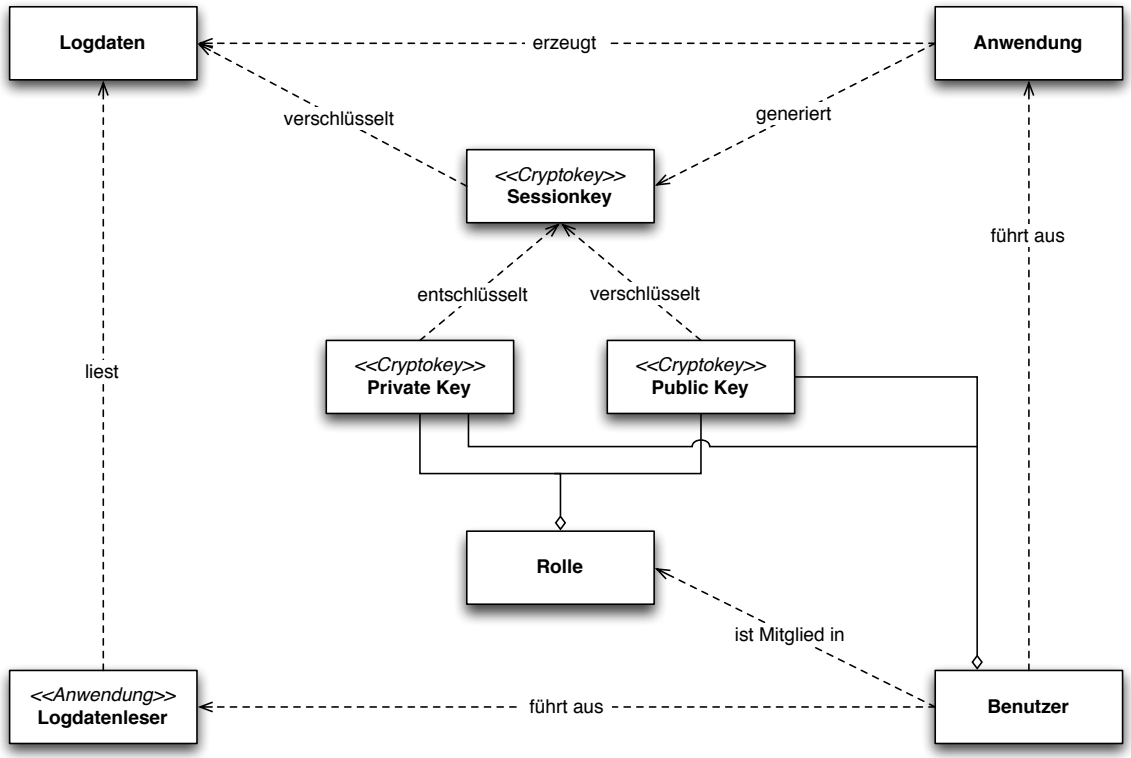

Abbildung 3.10: Klassendiagramm: Problembereichsmodell sicheres Protokollieren

```
Datum.crypt(Sessionkey), Datum.hash().crypt(PrivateKey), Sessionkey.crypt(PublicKey)
```

Dadurch ergeben sich weitere Möglichkeiten: Durch unterschiedliche Verschlüsselung des Sessionkey kann eine UND- bzw. ODER-Verschlüsselung erreicht werden; d.h. es kann erzwungen werden, dass zwei Besitzer von unterschiedlichen privaten Schlüsseln gemeinsam ein Datum entschlüsseln, es kann jedoch auch realisiert werden, dass einer der Schlüsselinhaber die Entschlüsselung vornehmen kann.

```
undCrypt(Sessionkey, PublicKey1, PublicKey2) :=
        Sessionkey.crypt(PublicKey1).crypt(PublicKey2)

oderCrypt(Sessionkey, PublicKey1, PublicKey2) :=
        (Sessionkey.crypt(PublicKey1), Sessionkey.crypt(PublicKey2)
```

Die Verschlüsselung für alle Rolleninhaber wird durch Verwaltung eines Schlüsselpaares für jede Rolle umgesetzt. Ein Problem ergibt sich dabei bei der Speicherung der Schlüsselpaare. Dies könnte auf einem gesonderten Schlüsselserver geschehen oder durch Verteilung der Rollenschlüssel an alle Mitglieder, wobei im Falle des Entzugs einer Mitgliedschaft alle Mitglieder mit neuen Schlüsseln ausgestattet werden müssten und alle Logdaten umzuverschlüsseln wären. Einfacher ist hier wieder eine ODER-Verschlüsslung des Rollenschlüssels in einem gesonderten Server, so dass jeder aktuelle Rolleninhaber in einer gesicherten Umgebung den Rollenschlüssel freigeben kann, der nach erfolgter kryptographischer Operation wieder gelöscht wird.

Die Pseudonymität kommt bei diesem Logging-Verfahren zum Tragen, wenn personenbezogene Daten z.B. nur über ein Vier-Augen-Prinzip oder unter geeigneten Auflagen nur für bestimmte Rollen zugänglich gemacht werden können. In der Anwendung werden diese Daten bereits verschlüsselt. Damit liegen Protokolldaten zwar vor, können jedoch nur über Einhaltung entsprechender Verfahren wieder zugänglich gemacht werden. Die Protokolldaten wären zu kategorisieren und entsprechend ihrer Sensibilität verschlüsselt aufzubewahren.

Kapitel 4

eXtreme Role-Engineering

> Sorry, I'm a bit of a stickler for paperwork. Where would we be, if we didn't follow the correct procedures?
>
> Brazil (film)

Das eXtreme Role-Engineering (xRE) wurde für die verteilte Administration einer Organisation mit dem Ziel entwickelt, Mitarbeitern ein strukturiertes Vorgehen für die Rollenadministration zur Verfügung zu stellen, das auch ohne Spezialwissen im Bereich RBAC einsetzbar ist. Typische Anwender des xRE sind Verwaltungsangestellte, Leiter von Instituten, Fachgebieten oder Fakultäten bzw. wissenschaftliche Angestellte, die im Auftrag die Rollenzuweisungen und Rollenverwaltungen vornehmen. Als Kontext wurde das im Kapitel 3 beschriebene System vorausgesetzt. Das entwickelte Verfahren ist allgemeiner einsetzbar, als durch die genannten Anforderungen definiert (siehe hierzu Kapitel 4.7). So kann xRE auch in anderen RBAC-Umgebungen und in unterschiedlichen Arbeitsumfeldern eingesetzt werden.

Wird ein RBAC-Modell konsequent verteilt administriert, so ergeben sich folgende Besonderheiten in Bezug auf die Rollenadministration:

- Die Menge der zu verwaltenden Rollenmitglieder ist vergleichsweise klein (typisch sind 10-100 Personen).
- Daraus folgt, dass die Zahl der für die Untereinheit zu verwaltenden Rollen typischer Weise zwischen 5 und 50 liegt.
- Die Rollenzuweisungen und Rollenverwaltungen werden von Administratoren durchgeführt, die hauptsächlich andere Aufgabenfelder besitzen. Typische Anwender sind also "Teilzeitadministratoren".
- Die Rollenadministratoren besitzen in der Regel sehr gute Kenntnisse über die Prozesse und die Personen in dem von ihnen verwalteten Umfeld.
- Änderungen am Rollenmodell der einzelnen Organisationseinheiten werden selten vorgenommen, z.B. bei Einführung neuer Dienste im System oder bei personellen Änderungen.

Daraus folgt, dass das Verfahren für die Administratoren leicht zu erlernen und in kurzen Zeitabschnitten durchführbar sein muss, ohne dabei die Qualität der resultierenden Rollenmodelle in Hinblick auf Konsistenz, Sicherheit und Transparenz negativ zu beeinflussen. Das xRE ist daher darauf ausgerichtet, den Benutzer über Dialogwerkzeuge zu unterstützen.

Das xRE Verfahren besteht aus einem Phasenmodell, ähnlich dem klassischen Role-Engineering, verbunden mit agilen Methoden, die dem eXtreme Programming entliehen sind. Folgende Phasen sind definiert (siehe Abb. 4.1):

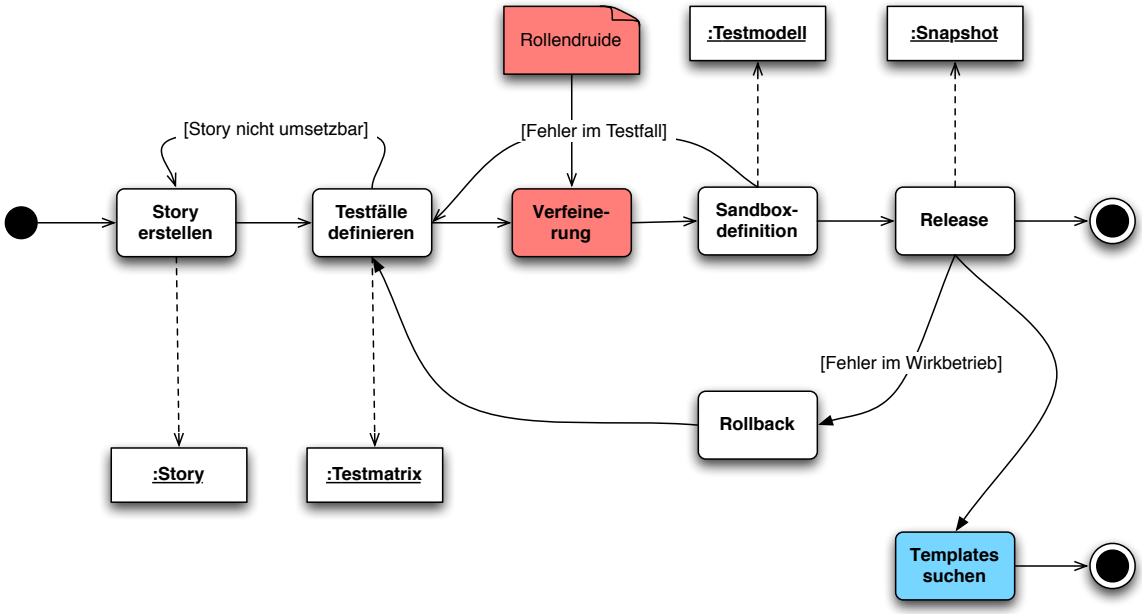

Abbildung 4.1: Zustandsdiagramm: Das eXtreme Role-Engineering Vorgehensmodell

Erstellung einer Story: Der Verwalter der Organisationseinheit beschreibt anhand von Beispielen, welche Anwendung für welche Personengruppe zugänglich gemacht werden soll. Ferner können Mitglieder einer Organisationseinheit Vorschläge für Stories dem Verwalter übermitteln.

Definition von Testfällen: Der Verwalter wählt geeignete Testpersonen aus seiner Einheit. Über eine GUI definiert er in einer Matrix, welche Personen welche Zugriffe auf die Anwendungen bekommen sollen.

Verfeinerung des Rollenmodells: Ein "Rollenfindungsdruide" genanntes Werkzeug ermittelt an Hand der Testfallspezifikation, ob eine geeignete Rolle oder ähnliche Rollen bereits im Modell vorhanden sind. In einem Dialog mit dem Benutzer wird das Rollenmodell verfeinert.

Rollenzuweisung in einer Sandbox: Sind die automatisierten Tests des Modells erfolgreich, wird das neue Modell in einer "Sandbox" installiert. Hier kann der Verwalter ein "Was wäre wenn"-Szenario durchtesten und damit logische Fehler aufspüren, die durch die automatisierten Tests verborgen blieben.

Release: Die in der Sandbox definierten Änderungen werden auf den Produktivserver kopiert. Die jeweiligen Zustände vor und nach der Änderung werden archiviert. Ein Rollback ist jederzeit möglich, falls in der Produktivumgebung Fehler auftauchen, die bei den Unit-Tests und in der Sandbox nicht aufgefallen sind.

Templates suchen: Ein Werkzeug durchsucht alle Organisationseinheiten nach ähnlichen Rollendefinitionen, um globale Vorlagen für häufig verwendete Rollen zu erstellen. Diese werden dem globalen Rollenverwalter vorgeschlagen und können wiederum im Dialog verfeinert werden.

Bei der Entwicklung von xRE mussten folgende Aspekte *nicht* berücksichtigt werden:

Anwendungsadministration: Das Hauptaugenmerk von xRE liegt bei der Rollenzuteilung und bei der Rollenadministration in Organisationseinheiten. Die Entwicklung von Anwendungen und die Definition geeigneter Anwendungsrollen wurden in das Verfahren nicht einbezogen.

Rollenhierarchien: Rollenhierarchien dienen der Strukturierung und Vereinfachung von Rollenmodellen. Auf der anderen Seite erhöhen Rollenhierarchien nicht nur die Komplexität der Modelle auf der Implementierungsseite, sondern auch auf Seiten der Rollenadministration. Da xRE auf eine kleine Zahl von Rollen ausgelegt ist, wurde von der Verwendung von Rollenhierarchien abgesehen.

Nebenbedingungen (Constraints): Nebenbedingungen wurden in RBAC-Systemen schon sehr früh vorgesehen. Jedoch existiert bislang keine allgemeine standardisierte Ausdrucksmöglichkeit. Lediglich für Spezialfälle, wie z.B. für zeitliche Nebenbedingungen wurden Sprachen entwickelt. Nebenbedingungen können gerade im Umfeld der Teilzeitadministratoren nur dann zum Einsatz kommen, wenn sie auf den jeweiligen Spezialfall zugeschnitten sind. Zur Entwicklung des Vorgehensmodells werden sie daher nicht betrachtet.

Die hier genannten Aspekte wurden bei der Entwicklung nicht berücksichtigt, werden jedoch durch die Verwendung von xRE nicht ausgeschlossen, wie in Kapitel 4.7 gezeigt wird.

Im Folgenden werden die funktionalen und nicht-funktionalen Anforderungen an das xRE Verfahren im Detail beschrieben. In den Abschnitten 4.2 und 4.3 folgt der Entwurf der Werkzeuge zur Unterstützung des Verfahrens. Betrachtungen zur Effizienz der Algorithmen werden in 4.4 diskutiert. Es schließen sich Überlegungen zur Fragestellung, wie Rollenmodelle mit Nebenbedingungen und Hierarchien unterstützt werden können an. Das Kapitel wird mit der Evaluation und Diskussion des Verfahrens abgeschlossen.

4.1 Anforderungen an das eXtreme Role-Engineering

Das xRE System umfasst sieben Anwendungsfälle für fünf Benutzergruppen:

Wir unterscheiden im Folgenden die Rollenadministratoren mit Zugriff auf globale Aspekte des Rollenmodells und vorausgesetztem Spezialwissen auf dem Gebiet RBAC und den Strukturverwaltern, die dem in der Einleitung dieses Kapitels beschriebenen Profil von Teilzeitadministratoren entsprechen. Ferner sind Anwendungsintegratoren und Anwendungsverwalter zu betrachten, die für die Bereitstellung von Diensten und Rechten verantwortlich sind. Als Benutzer werden die vormals genannten sowie alle sonstigen Anwender bezeichnet, die über das RBAC-Modell mit Rechten versorgt werden.

Für diese Benutzergruppen liegen folgende Anwendungsfälle vor:

Initialisierung des Modells: Herstellung eines definierten Ausgangszustands durch die Rollenadministratoren. Automatische oder manuelle Erstellung von Testmatrixen, die den initialen Zustand als konsistent identifizieren.

Vorlagen verwalten: Das Anlegen, Ändern und Löschen von Rollenvorlagen durch die Rollenadministratoren.

Anwendungsfall beschreiben: Jeder Benutzer des Systems soll in die Lage versetzt werden, ein Szenario zu beschreiben, das dann durch den Strukturverwalter im RBAC-Modell umgesetzt werden soll. Dabei kann es sich um das Hinzufügen, Ändern oder Löschen von Rechten für bestimmte Personengruppen handeln. Die Entscheidung über die Umsetzung der Szenarien treffen Strukturverwalter.

Anwendungen verwalten: Anwendungsintegratoren müssen in die Lage versetzt werden, Anwendungen hinzuzufügen, zu ändern und zu löschen und damit einen Dienst für das RBAC-Modell zur Verfügung zu stellen.

Anwendungsfälle restrukturieren: Anwendungsverwalter müssen in die Lage versetzt werden, Mengen von Rechten für die jeweiligen Dienste zu definieren. Mit der Änderung von Rechten sind jeweils Restrukturierungen der damit verbundenen Szenarien verbunden. Die Strukturverwalter müssen auf diese Strukturierung entsprechend der Anforderungen in ihren Organisationseinheiten reagieren.

Rollen restrukturieren: Die Restrukturierung eines RBAC-Teilmodells nutzt den Anwendungsfall "Anwendungsfall beschreiben" und bildet den Hauptteil des in Abb. 4.1 dargestellten Vorgehensmodells (In der Verfeinerung der Anwendungsfälle werden die Zusammenhänge zwischen Anwendungsfällen und Vorgehensmodell deutlicher.).

Rollenmitgliedschaft verwalten: Nach der Restrukturierung des Rollenmodells, d.h. der Definition von Rollen und der Ausstattung mit Rechten, müssen die Mitgliedschaften, also die Zuweisungen zwischen Benutzer und Rollen, definiert werden können. Dies geschieht ebenfalls durch den Strukturverwalter.

Verfeinert man die hier aufgestellten Anwendungsfälle, stellt man fest, dass zwischen den Anwendungsfällen komplexe Abhängigkeiten bestehen. Nach dem Löschen einer Rolle müssen beispielsweise alle Szenarien überarbeitet werden, die von dieser Rolle abhängig sind. Die mit der Rolle verbundenen Rechte müssen ggf. wieder an andere Rollen gebunden werden, um die damit verknüpften Dienste weiter nutzbar zu machen. Ändert sich ein Dienst und die damit verbundenen Rechte für die Anwendungen, so müssen in den Struktureinheiten wiederum Szenarien überarbeitet werden und damit Rollenstrukturierungen ausgelöst werden.

Diese Zusammenhänge wurden bereits untersucht: [29] stellt die möglichen Anwendungsfälle und die daraus entstehenden Implikationen bei der Verwaltung von Rollen zusammen. Betrachtet werden Szenarien, Geschäftsrollen und Rollenmitglieder, Anwendungsrollen (in [29] Permissions) und die Verknüpfungen zwischen Geschäftsrollen und Anwendungsrollen (in [29] PA = Permission Assignment). Diese Zusammenhänge werden bei der folgenden Verfeinerung der Anforderungen berücksichtigt.

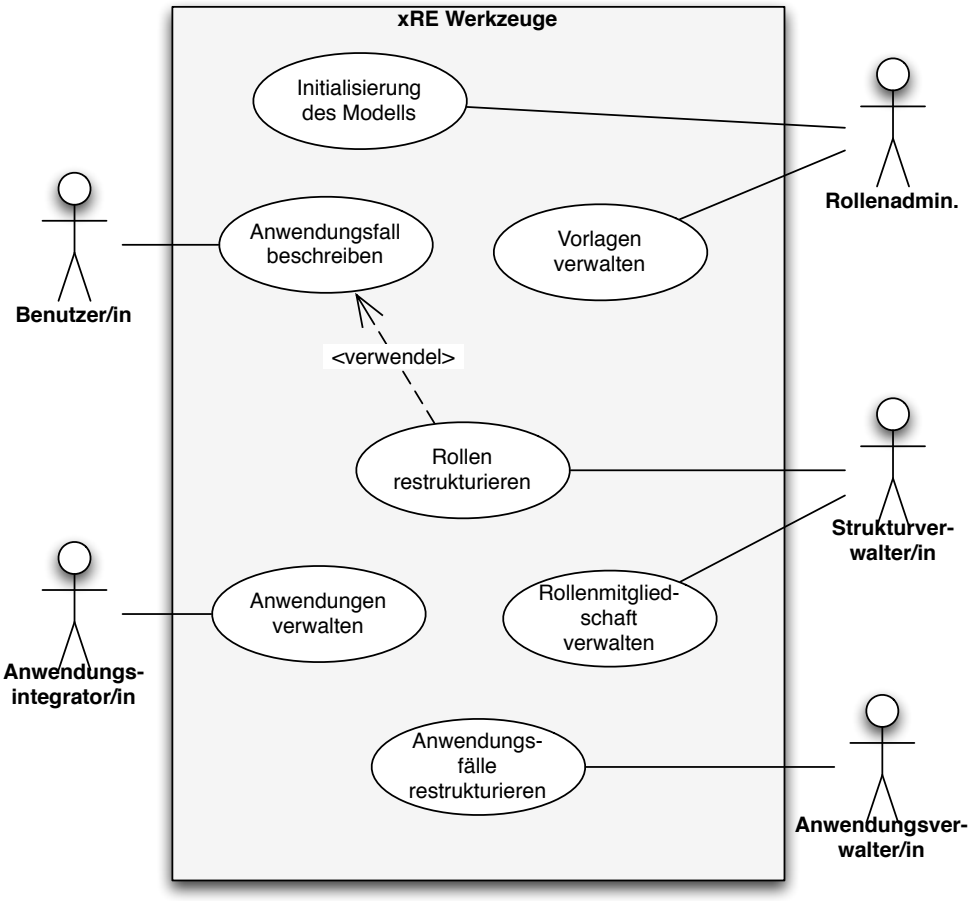

Abbildung 4.2: Anwendungsfalldiagramm: xRE Übersicht

4.1.1 Funktionale Anforderungen an die globale Rollenadministration

Die Initialisierung des Modells enthält minimal die Einrichtung mindestens einer Struktureinheit und die Verteilung der Rollen "Strukturverwalter", "Anwendungsverwalter", "Anwendungsintegrator" und "Benutzer", wobei die Bekleidung mehrerer Rollen durch eine Person nicht ausgeschlossen ist.

Rollenadministratoren müssen ferner in die Lage versetzt werden, Rollenvorlagen zu definieren, die dann in die Vorschlagsliste für die Strukturverwalter aufgenommen werden. Die xRE Werkzeuge erstellen Vorschläge für die Erstellung von Vorlagen durch Vergleich von Rollendefinitionen der verschiedenen Organisationseinheiten. Definieren mehrere Strukturverwalter dieselben oder sehr ähnliche Rollen, kann es sinnvoll sein, eine solche Rolle bereits als Vorlage für weitere Organisationseinheiten anzubieten. Die Entscheidungen hierüber trifft der Rollenadministrator, der mit der Vorlagenerstellung ferner in die Lage versetzt wird, die Strukturverwalter in Richtung vorausschauender Definition von Rollen zu beeinflussen.

4.1.2 Funktionale Anforderungen für die Strukturverwaltung

Die Strukturverwaltung umfasst die beiden durch die Strukturverwalter genutzten Anwendungsfälle "Rollen restrukturieren" und "Rollenmitgliedschaft verwalten". Dabei stellt "Rollenmitgliedschaft verwalten" eine Erweiterung des Anwendungsfalls "Rollen restrukturieren" dar. "Rollen restrukturieren" umfasst weiterhin die Funktionen "Rollen hinzufügen", "Rollen löschen", "Rollen teilen" und "Rollen zusammenfassen". Für die Restrukturierung werden gemäß Vorgehensmodell in Abb. 4.1 Anwendungsfälle beschrieben und Testfälle erstellt. Ferner ist es möglich, das Rollenmodell zu testen. Die Zusammenhänge zwischen den einzelnen Anwendungsfällen sind in Abb. 4.3 dargestellt.

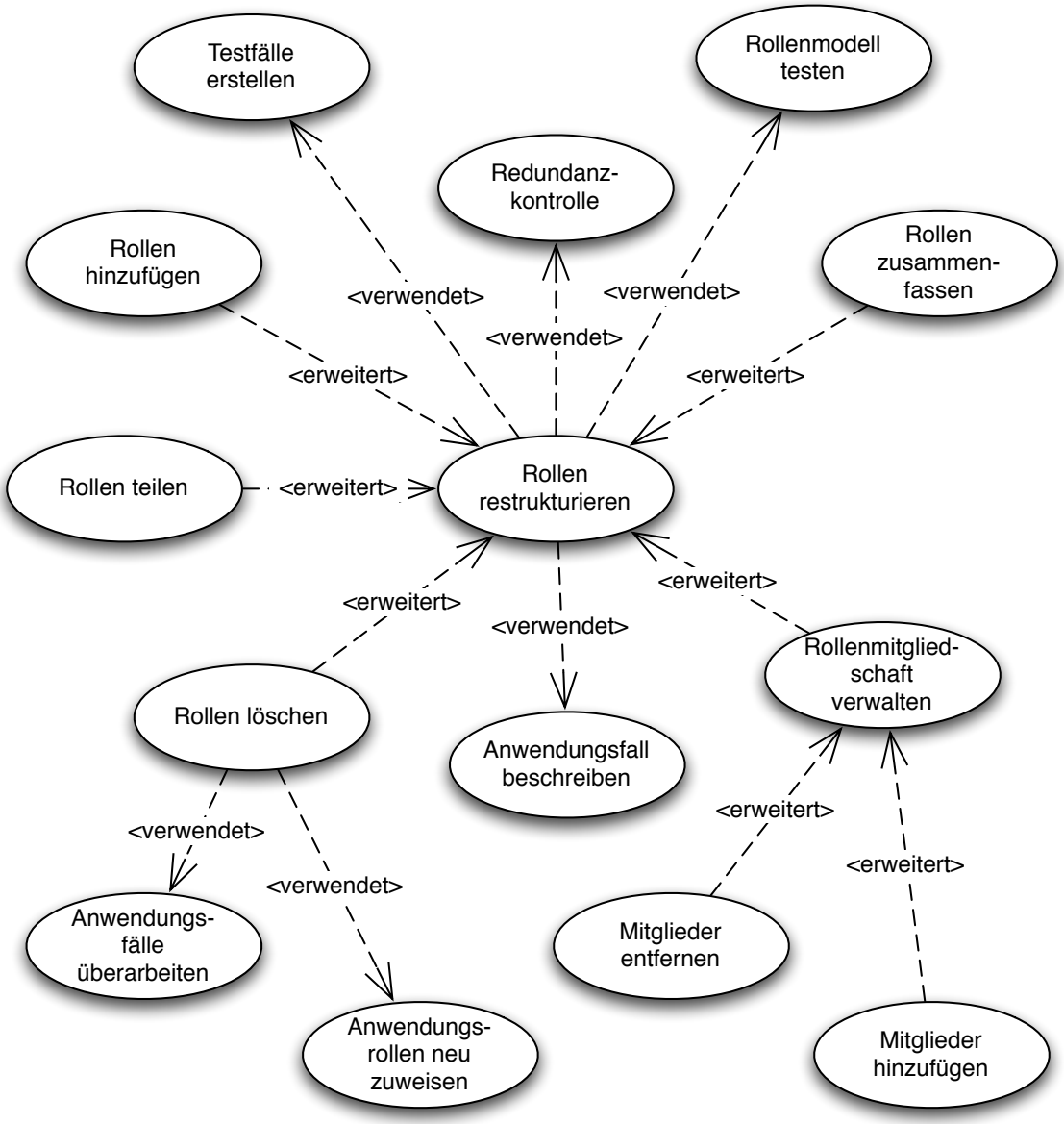

Abbildung 4.3: Anwendungsfalldiagramm: Verfeinerung: Rollen restrukturieren

Das System muss sicherstellen, dass jeweils alle von einer Aktion abhängigen Folgeaktionen abgearbeitet wurden, bevor das resultierende Teilmodell getestet und schließlich produktiv geschaltet werden kann. Damit wird verhindert, dass inkonsistente Teilmodelle in den produktiven Einsatz kommen. Der Anwender muss im Dialog darüber informiert werden, welche Entscheidungen jeweils Folgeaktionen implizieren und wie viele offene Arbeitsschritte existieren, um das aktuell bearbeitete Szenario abzuarbeiten. Ein aktueller Zustand muss gespeichert und zu einem späteren Zeitpunkt wieder aufgenommen werden können.

4.1.3 Verwaltung von Anwendungen

Die Verwaltung der Anwendungen selbst ist nicht Bestandteil des hier zu entwerfenden Systems. Änderungen an den Anwendungen und damit an den Rechten bzw. Anwendungsrollen haben jedoch Einfluss auf die Rollendefinitionen der Struktureinheiten. Die Anwendungsfälle mit Auswirkung auf die Teilmodelle der Organisationseinheiten müssen daher betrachtet werden.

Die Anwendungsverwaltung besteht aus den beiden großen Bereichen: "Anwendungen verwalten", hierzu gehört das Hinzufügen und Löschen von Anwendungen sowie "Anwendungsrollen verwalten". Wie in Abb. 4.2 gezeigt, können diese Aufgaben auf unterschiedliche Rollen verteilt werden. Anwendungen werden im Vergleich zu anderen Anwendungsfällen selten hinzugefügt oder gelöscht. Daher muss auch die Rolle des Anwendungsverwalters nur selten aktiviert werden. Zur Wartung der Anwendungen gehört auch die Anpassung des Funktionsumfanges an die aktuellen Gegebenheiten und Prozesse. Solche Anpassungen ziehen eine Anpassung der Anwendungsrollen mit sich. Für die geänderte Funktionalität müssen die Rechte verwaltet werden können. Um die neuen Funktionen in den Organisationseinheiten verfügbar zu machen, ist ein Eingreifen der Strukturverwalter notwendig. Diese müssen Personen ihrer Organisationseinheit mit den Rechten zur Nutzung der neuen Funktionen ausstatten. Dieser Prozess wird durch das Generieren eines Szenario-Vorschlags ausgelöst. Aus Sicht der/des Anwendungsverwalter/s wird eine Verwaltung der Anwendungsrollen mit der Beschreibung eines neuen Szenarios abgeschlossen. Es ist die Frage zu beantworten, welche neuen Funktionen nach den Anpassungen zur Verfügung stehen oder welche ggf. eingestellt oder geändert wurden. Die Strukturverwalter werden über diese Änderungen informiert, indem diese als neues Wunsch-Szenario zur Verfügung gestellt wird. Das Szenario kann dann an die Anforderungen der jeweiligen Einheit angepasst werden. Mit diesem Szenario wird dann der xRE-Prozess angestoßen.

Abb. 4.4 zeigt eine Verfeinerung der Anwendungsfälle der Anwendungsverwaltung und die Schnittstelle zur Strukturverwaltung über den Fall "Anwendungsfall beschreiben". Die Beschreibung der Anwendungsfälle dient ferner der Dokumentation im System. Über diese Szenario-Beschreibungen kann später nachvollzogen werden, welcher Zweck mit den Änderungen am Teilmodell verfolgt wurde.

4.1.4 Anforderungen aus der Anwendung von XP-Methoden

Wie in der Einleitung von Kapitel 4 skizziert, besteht der Lösungsansatz von xRE u.a. in der Anwendung der Prinzipien des XP (vergl. Kapitel 2.5.1) auf das Role-Engineering. Daraus ergeben sich die folgenden Anforderungen an die Methode.

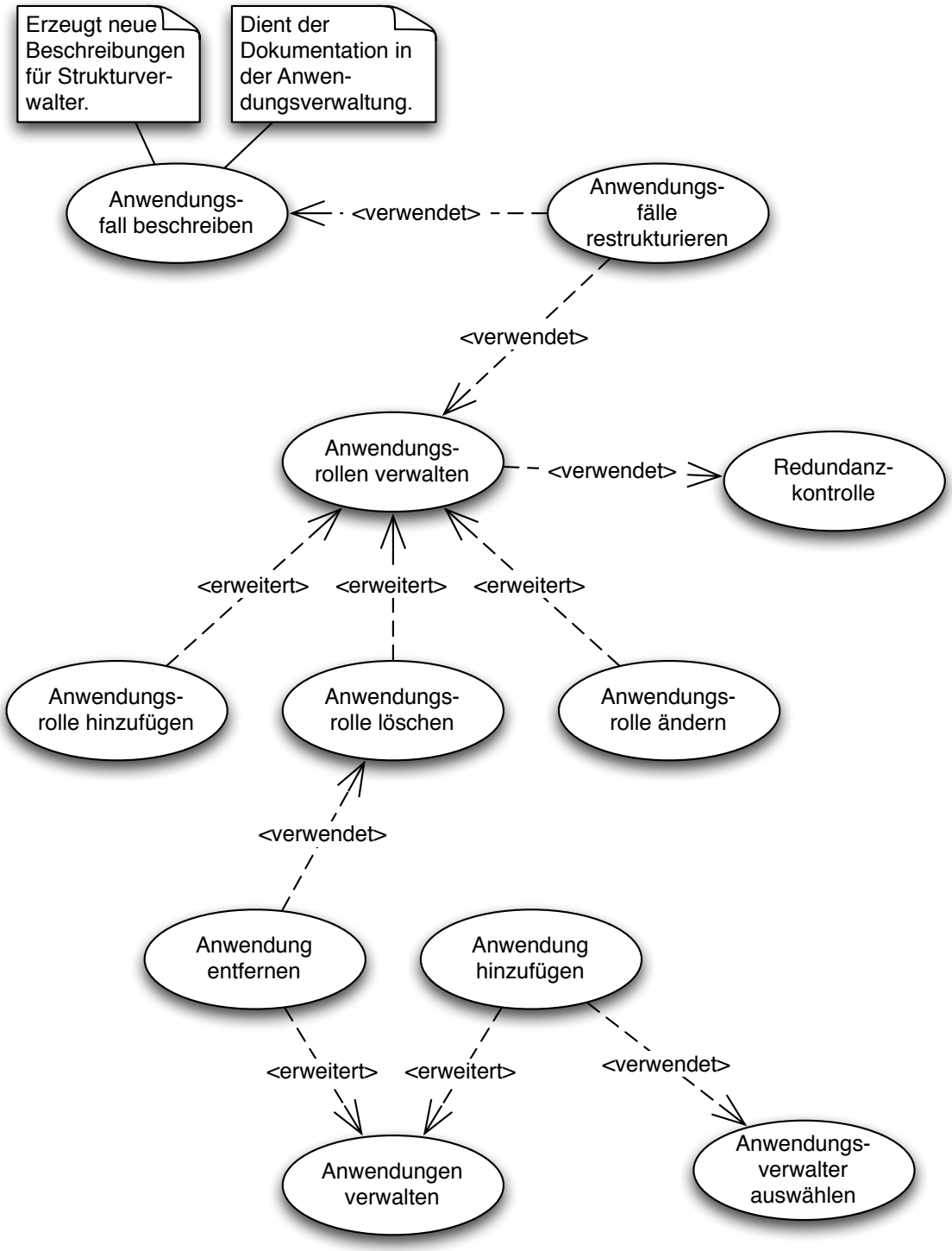

Abbildung 4.4: Anwendungsfalldiagramm: Verfeinerung: Anwendungsverwaltung

Kleine Releases

Die Komplexität eines Modells, unabhängig davon, ob es sich um das Modell eines Softwaresystems oder um ein Rollenmodell handelt, wächst mit seiner Größe stark an. Aus der Softwareentwicklung weiß man, dass kleine, sehr einfach gehaltene Projekte tendenziell eher zum Erfolg führen, als große und damit komplexe [10]. Die Idee, die auch beim XP zum Tragen kommt, besteht darin, das große Projekt in eine geeignete Zahl kleiner Teilprojekte zu zerlegen. Im Gegensatz zu klassischen Methoden, die ein Zerlegen in verschiedene Iterationsschritte und Meilensteine kennen, wird jedoch jeder Teil für sich betrachtet und auch bei der Planung jeweils nur vom nächsten Teilerfolg ausgegangen. Nur so wird tatsächlich Komplexität reduziert.

Die kleinen Releases können direkt auf das xRE übertragen werden. In einem Release werden beim xRE jeweils nur kleine Personengruppen ausschließlich einem Anwendungsszenario zugeführt. Die Zugriffsmodellierung wird dabei jeweils in den kleinen Schritten mitgeführt. So entsteht das Rollenmodell in kleinen Abschnitten, die einzeln für sich getestet werden können.

Planungsspiel

Beim XP wird einer Iteration ein Planungsspiel vorangestellt, das Auskunft darüber geben soll, was nach einer Iteration möglich sein soll. Dabei dient das Planungsspiel auch der Auswahl der Funktionalität, die in dieser oder erst in einer der nächsten Iterationen implementiert werden soll.

Das xRE benutzt die Planungsspiele analog. Auch hier soll beschrieben werden, was nach Implementierung im Rollensystem möglich sein soll. Es liegt in der Natur der Sache, dass hierbei ein Schwerpunkt darauf liegt, zu beschreiben, *wer* jeweils in welche Lage versetzt werden soll oder was sich am aktuellen Zustand ändern soll. Die "Stories" werden beim xRE archiviert und stellen einen Teil der Dokumentation darüber dar, die nachweist, warum wann welche Änderung durchgeführt wurde.

Stories müssen nicht unbedingt vom Rollenadministrator geschrieben werden. Sie können auch über ein Workflowsystem zur Bearbeitung vorgelegt werden. So kann eine Anforderung für die Änderung am Rollenmodell vom Vorgesetzten kommen, durch die Implementierung eines neuen Dienstes im Softwaresystem oder durch Wünsche der Anwender. Abhängig von der Menge dieser Änderungswünsche und von der Arbeitskultur in der Organisationseinheit kann es sinnvoll sein, dass sich alle Beteiligten treffen, um die Stories zu besprechen und zu priorisieren. Dabei können Stories ggf. konkretisiert oder in mehrere Stories zerlegt werden.

Bei der Implementierung der xRE-Werkzeuge ist darauf zu achten, dass das Planungsspiel in das Änderungsverwaltungssystem einfließt.

Tests

Die Tests ersetzen beim XP die Spezifikation von Softwaremodulen. Die Tests der Module werden jeweils zuerst entworfen. Danach erst folgt die Implementierung, die dann gegen die vorab entworfenen Tests validiert werden kann.

Während die Erstellung aussagekräftiger Unit-Tests in der Softwareentwicklung eine der größten Herausforderungen darstellt, lassen sich Unit-Tests im Bereich der Zugriffskontrolle stark vereinfachen. So kann ein Test als Matrix dargestellt werden, bei der eine Zeile jeweils einer Testperson zugeordnet ist, wogegen die Spalten jeweils einem Zugriffsrecht entsprechen.

Die Abstraktion, die bei der rollenbasierten Zugriffskontrolle den entscheidenden Effizienzvorteil bringt, gibt man zur Erstellung der Unit-Tests auf. Im ersten Schritt muss der Rollenadministrator eine geeignete Auswahl an Testpersonen wählen. Danach wählt er eine geeignete Menge an Rechten für das gegebene Szenario. In einem weiteren Schritt wird dann markiert, welche der Testpersonen am Ende welche Zugriffe durchführen können sollten.

Das Unit-Testing ermöglicht auch das Testen von Nebenbedingungen (siehe Kapitel 4.5). Diese werden als Wahrheitsausdruck in die Zellen der Matrix geschrieben. An die Stelle eines Wahrheitswertes (`true` ≡ "Person hat das Recht", `false` ≡ "Recht steht der Person nicht zur Verfügung") wird ein logischer Ausdruck eingesetzt.

Die xRE-Unit-Tests werden wie die Planungsspiele archiviert. Bei jeder Änderung am Rollenmodell werden alle Unit-Tests erneut durchgeführt. Schlägt ein Test fehl, wird dieser dem Rollenadministrator zusammen mit der Story präsentiert, für die er entworfen wurde und einer Übersicht zur daraus resultierten Rollendefinition. Am Ende darf der Rollenadministrator entscheiden, ob es ein gewünschter Effekt ist, dass der Test nun fehlschlägt und ob der Test zukünftig angepasst durchlaufen werden soll. Die Entscheidung ist dann jeweils wieder zu kommentieren.

Systemmetapher

An die Stelle der Systemmetaphern beim XP tritt im xRE-Verfahren die Rollenmetapher. Bei der Abstraktion der Zugriffsrechte über Rollenbezeichner, Rollenmitglieder und an die Rollen gebundenen Rechte und Pflichte handelt es sich bereits um eine Metapher, die Entwickler, wie Nutzer gemeinsam benutzen. Aus diesem Grund führt xRE keine weiteren Metaphern ein.

Einfacher Entwurf

XP wählt jeweils die einfachste Implementierung, die funktionieren kann. Dieses Prinzip soll auch beim xRE verfolgt werden. Dabei ist darauf zu achten, dass nicht nur die Implementierung, d.h. die Benutzer–Rollen und die Rollen–Rechte Zuordnung einfach zu halten sind. Ein weiterer wichtiger Punkt ist die Verständlichkeit der Metaphern oder anders gesagt: Die verständliche Bezeichnung der Rollennamen. Eine Rolle sollte eine Entsprechung aus der Arbeitswelt widerspiegeln und nicht ausschließlich aus Überlegungen der effizienten Benutzer–Rollen–Rechte Definition entstehen.

"Einfachheit" besitzt im xRE die beiden Dimensionen:

1. Einfache Implementierung der Rolle, z.B. durch geeignete Wiederverwendung von bereits definierten Rollen.

2. Verständliche Bezeichnung und intuitiver Umgang mit Rollendefinitionen durch sprechende Bezeichner, die eine möglichst exakte Entsprechung in der Arbeitswelt besitzen.

In diesem Spannungsfeld sind die jeweiligen Entwurfsentscheidungen für das Rollenmodell zu treffen.

Refaktorisierung

Die Refaktorisierung ist ein notwendige Maßnahme, die in Kauf genommen werden muss, wenn man in kleinen Iterationsschritten vorgehen will und auf eine umfassende Planung verzichtet. Es wird jeweils implementiert, was zur nächsten Iteration nötig ist. Damit sind fast

zwangsweise Änderungen zur nächsten Iteration nötig, die eine widerspruchslose Implementierung der Stories der letzten und der aktuellen Story zulassen. Das bereits "Fertige" muss immer wieder "angefasst" und weiterentwickelt werden.

Um Fehler bei der Refaktorisierung zu vermeiden, werden nach dem Refaktorisierungsprozess erneut die Unit-Tests angestoßen, um zu prüfen, ob bei der Weiterführung der Arbeit neue Fehler aufgetaucht sind.

Kaum ein anderer Implementierungsprozess ist anspruchsvoller für den Implementierer, als die Refaktorisierung. Im Bereich der Software-Entwicklung stehen daher viele Werkzeuge für diesen Prozess zur Verfügung, die dabei helfen, Übertragungs- oder Flüchtigkeitsfehler zu reduzieren und den bestehenden Code auf unterschiedliche Weise zu analysieren.

Die reichhaltige Unterstützung des Refaktorisierungsprozesses beim xRE durch Softwarewerkzeuge ist daher unbedingt anzustreben!

Programmierung in Paaren

Die Programmierung in Paaren wird beim XP zum einen zur Verteilung des Wissens über das Projekt und zum anderen zur Erhöhung der Quellcodequalität eingesetzt. Die Modellierung in Paaren ist auch für das xRE wünschenswert. Im Gegensatz zur Paarprogrammierung ist die Modellierung in Paaren leichter zu organisieren. Eine Rollenmodelländerung wird selten vorkommen und eine komplette xRE-Iteration ist in der Regel in wenigen Stunden oder sogar Minuten durchführbar. Auch vor der Einführung des xRE-Verfahrens war es nach Erfahrungsberichten bei einigen Benutzern üblich, sich zu zweit zur Rollenmodellierung zusammenzusetzen. Dabei behielt ein Modellierer den Modellierungsweg im Auge, während sich der andere Modellierer um die technische Umsetzung kümmerte.

Je nach Komplexität der zu verwaltenden Rollenmodelle ist es möglich, auf die Paarmodellierung zu verzichten. Gerade wegen des im Gegensatz zum XP viel geringeren Synchronisationsaufwands ist die Paarmodellierung in jedem Fall zu empfehlen. Geeignete Paare können Rollenverwalter und Stellvertreter sein, aber z.B. auch, wie an der TU Berlin üblich, Sekretariate und Professoren. Da es für die Rollenverwaltung *unbedingt* eine Stellvertreterregelung geben sollte, kann die Paarmodellierung dazu genutzt werden, um Rollenverwalter und Stellvertreter jeweils auf dem aktuellen Stand zu halten.

Gemeinsames Code-Eigentum

Es scheint im Allgemeinen leichter zu fallen, das Rollenmodell als gemeinsames Eigentum anzuerkennen, als Programmcode. Das liegt vermutlich an der Schaffungshöhe und dem damit verbundenen "gefühlten" Urheberrecht des initialen Programmierers. Die Rollenmodellierung muss in jedem Fall als gemeinsames Eigentum der Organisation bzw. der jeweiligen Untereinheit akzeptiert werden. Zum einen ist die Verantwortung für eine Änderung zu verstehen und zu akzeptieren, zum anderen muss in der Veränderbarkeit des Modells auch der Schlüssel zur Flexibilität verstanden werden.

Kontinuierliche Code-Integration

Bei der Entwicklung des xRE-Vorgehens wurde von einem bereits im Einsatz befindlichen Zugriffskontrollmodell ausgegangen. Die Änderungen am Rollenmodell sollen jeweils am Ende einer Iteration zum Einsatz kommen, also wie beim XP kontinuierlich integriert werden. Vor

jeder neuen Iteration wird jeweils ein Abbild des aktuellen Zustands gemacht, so dass im Fehlerfall wieder auf den letzten wohl definierten Zustand zurückgesprungen werden kann.

40-Stunden-Woche

Wie bereits an anderer Stelle beschrieben, handelt es sich beim Rollenmodellierungsprozess um keine Tätigkeit, die in der Regel mehrere Tage in der Woche in Anspruch nimmt. Auf der anderen Seite ist es nicht empfehlenswert, den xRE-Prozess am Ende eines harten Arbeitstages mit vielen Überstunden durchzuführen. Die Verabredung für eine Modellierung in Paaren kann so auch helfen, einen Zeitpunkt zu finden, bei dem die Konzentration noch gut genug ist, um keine Fehlentscheidungen zu treffen.

Kundenvertreter im Team

Die xRE-Modellierung stellt im Idealfall eine Art von Prosumer-Tätigkeit dar; d.h. der Modellierer ist nicht nur der Implementierer der RBAC-Teilmodelle ("producer"), sondern auch Nutzer des Modells ("consumer"). In Fällen, in denen der Rollenadministrator unsicher in der Benutzung der zu verwaltenden Anwendungen ist, weil er beispielsweise selbst nicht darauf geschult ist, sollte mindestens ein vertrauenswürdiger Vertreter der späteren Nutzer und ggf. auch ein Vertreter der Anwendungsbetreiber hinzugezogen werden, um Zweifelsfälle klären zu können. Erfahrungsgemäß lassen sich viele Fragen zur Rechtevergabe auch spontan telefonisch klären. Hierzu ist es jedoch erforderlich, dass zu den jeweiligen Rechten, die vergeben werden können, Ansprechpartner definiert sind, die auch telefonisch oder per E-Mail erreichbar sind. Wenn nötig, kann die Modellierung zunächst vorbereitet werden und das Release erst nach Klärung der letzten Fragen freigegeben werden.

Programmierrichtlinien

Die Programmierrichtlinien helfen beim XP, den Programmcode für alle lesbarer zu machen und nehmen bei unterschiedlichen Umsetzungsmöglichkeiten die Entscheidung ab. Die Möglichkeiten bei der Modellierung sind bei weitem nicht so umfassend, wie bei einer Implementierung in einer höheren Programmiersprache. Trotzdem bietet es sich auch hier an, Richtlinien festzulegen. Festzulegen sind z.B.:

- Regeln zur Namenvergabe von Rollen, d.h. wie werden Rollen betitelt (männliche, weibliche, neutrale Form, Sichtweise bei der Namensvergabe, also aus Sicht des Softwaresystems oder aus Sicht der Anwender etc.).
- Ein "Best-Practice" zum Zusammenfassen oder Zerlegen von Rollen, d.h. wann werden aus einer Rolle zwei gemacht, um die Flexibilität zu erhöhen und wann werden Rollen zusammengelegt, um die Menge der Rollen übersichtlich zu halten?
- Richtlinien zur Entscheidung über Mitgliedschaften in Rollen. Wann werden einer bestehenden Rolle weitere Rechte hinzugefügt und wann wird eine neue Rolle erzeugt, die ggf. wieder dieselben Mitglieder enthält oder wird eine weitere, abstrakte Rolle geschaffen, die dann Mitglied in den zwei Rollen wird?

Die genannten Fragestellungen dienen im Rollenmodell lediglich des leichteren Verständnisses *auch* durch dritte. Tatsächlich ist es möglich, äquivalente Modelle zu erstellen, die der

einen oder anderen Richtlinie folgen. Sicherheitsrelevante Richtlinien sollten nach Möglichkeit über Nebenbedingungen fest in das Rollenmodell integriert werden. Hierzu gehören z.B. Vertreterregelungen oder bestimmte sich ausschließende Rollen und Rechte. Was aus technischen Gründen nicht als Nebenbedingung implementiert werden soll, kann selbstverständlich auch in ein schriftlich fixiertes Regelwerk für Rollenadministratoren aufgenommen werden. Eine organisatorische Lösung ist an dieser Stelle evtl. fehleranfälliger, bietet jedoch das Potential, nicht betrachtete Ausnahmen auf der organisatorischen Ebene zu berücksichtigen.

Tabelle 4.1: Übersicht über die, durch die aus dem XP adaptierten Methoden, abgeleiteten Anforderungen

XP	**xRE**
Kleine Releases	Modellierung von einzelnen Anwendungsfällen, Aufbau des Modells in kleinen Schritten
Planungsspiel	Anforderungsanalyse wird durch Beschreibung von Szenarien (Story) realisiert, Stories können priorisiert werden.
Tests	Aufbau einer Test-Zugriffsmatrix, gegen die das Modell getestet wird
Systemmetapher	An Stelle der Systemmetaphern treten die Rollenbezeichner
Einfacher Entwurf	Einfachheit durch Wiederverwendung vs. intuitiv verständliche Rollennamen
Refaktorisierung	Prinzip der jeweils einfachsten Implementierung mit Notwendigkeit der stetigen Refaktorisierung
Paarprogrammierung	Die Modellierung in Paaren wird empfohlen, ist aber aber nicht zwingend, weil das Rollenmodell in der Regel nicht die Komplexität von Programmcode besitzt.
Gemeinsames Code-Eigentum	Das Rollenmodell gehört der Organisation / Organisationseinheit. Einschränkungen dieses Paradigmas bestehen durch die Zugriffsrechte auf dem jeweiligen Teilmodell.
Kontinuierliche Code-Integration	Jedes Release wird zunächst in das Wirksystem eingepflegt und dort angewendet.
40-Stunden-Woche	Hat keine große Bedeutung, da Rollenmodellierung kein Full-Time-Job ist.
Kundenvertreter im Team	Entweder ist Rollenverwalter Prosumer oder Vertreter müssen hinzugezogen werden.
Programmierrichtlinien	Richtlinien zur Modellierung müssen definiert und schriftlich fixiert werden.

4.1.5 Nichtfunktionale Anforderungen

Im Folgenden werden die Anforderungen aufgeführt, die durch die Systemumgebung vorgegeben werden. Dies umfasst nicht nur technische, sondern auch organisatorische und rechtliche Aspekte.

Die xRE Werkzeuge nutzen den RBAC Controller, um auf die jeweiligen RBAC Teilmodelle zuzugreifen. Aus architektonischer Sicht handelt es sich bei den xRE Werkzeugen um neu definierte "Views" einer Model-View-Controller Architektur. Über die Werkzeuge werden die im Rollenmodell enthaltenen Geschäftsrollen administriert. Informationen über die Struktureinheiten werden für die Definition der Gültigkeitsbereiche benötigt. Nutzt ein Strukturverwalter die xRE Werkzeuge, so wird ermittelt, für welche Einheiten er zuständig ist. Bei der Bearbeitung der Geschäftsrollen wird jeweils der Ausschnitt zur Verfügung gestellt, der an die jeweilige Einheit gebunden ist (siehe Abb. 4.5).

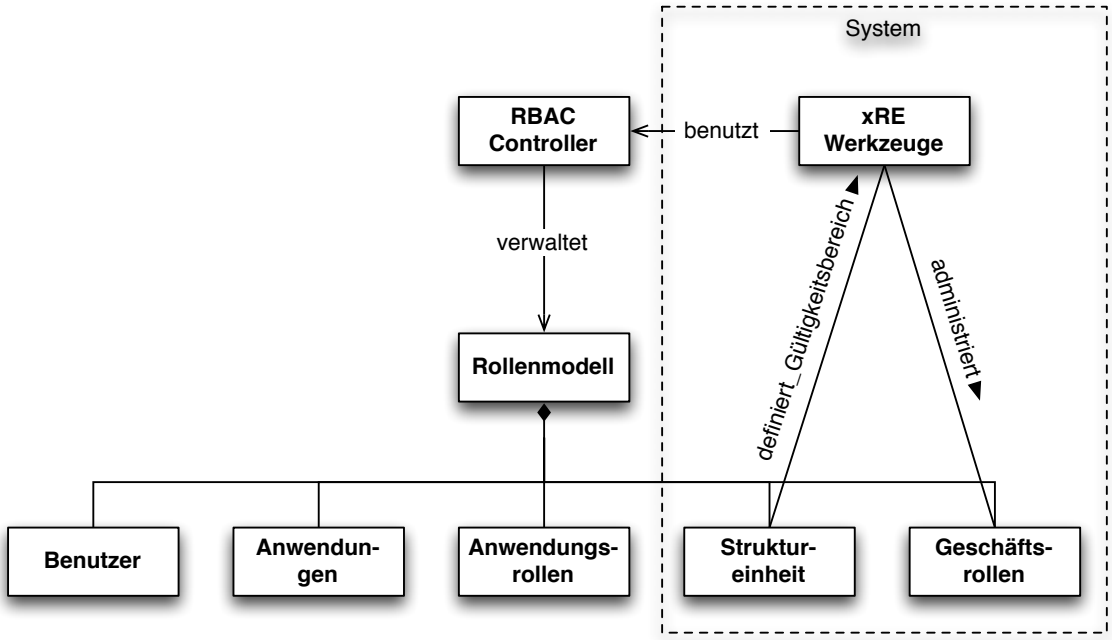

Abbildung 4.5: Klassendiagramm: Systemgrenzen der xRE Werkzeuge

Das zu entwerfende System muss in die in Kapitel 3 beschriebene Umgebung integriert werden können. Es *sollten* so wenig Annahmen wie möglich in Bezug auf das Rollenmodell und den verwendeten RBAC Controller gemacht werden, um das Verfahren möglichst universell einsetzbar zu gestalten.

4.2 Entwurf des eXtreme Role-Engineering Verfahrens

eXtreme Role-Engineering ist als Role-Engineering Verfahren mit Softwareunterstützung entworfen. Das Verfahren selbst kann auch ohne Softwareunterstützung angewendet werden. Es umfasst die Phasen "Story erstellen", "Testfälle definieren", "Verfeinerung" der Rollen,

"Sandboxdefinition" und "Release". Ferner werden Rollenadministratoren durch eine "Templatesuche" unterstützt (siehe Abb. 4.1 auf Seite 62).

Die Phasen werden jeweils durch eigene Werkzeuge unterstützt (siehe Tabell 4.2).

Tabelle 4.2: Phasen des xRE Verfahrens mit den jeweiligen Werkzeugen zur Unterstützung

xRE-Phase	**Werkzeug**
Story erstellen	Storyboard
Testfälle definieren	xRE UNIT
Verfeinerung	Rollenfindungsdruide
Sandboxdefinition	Sandbox
Release	kein eigenes Werkzeug
alle Phasen	xRE Verwalter
alle Phasen	xRE Refactoring Werkzeuge

Im Folgenden werden die Algorithmen und Datenstrukturen entworfen, die zur Anwendung des Verfahrens und für den Entwurf der Werkzeuge nötig sind.

4.2.1 Rollenähnlichkeit

Das Finden von ähnlichen Rollen spielt im Rahmen der Softwarewerkzeuge für xRE eine wesentliche Rolle. So können durch die Feststellung von Ähnlichkeiten ungewollte Redundanzen vermieden werden, aber auch Vorlagen aus häufig konfigurierten Kombinationen deduziert werden. Das Finden von geeigneten Modellen zur Ermittlung ähnlicher Rollen ist ein wichtiges Ziel dieser Arbeit. Im Folgenden werden daher verschiedene Varianten gezeigt und die Auswahl eines Modells begründet.

Voraussetzungen

Gegeben sind Rollen R und Permissions P sowie eine Rolle-Permission-Zuordnung (Permission Assignment) $PA \subseteq P \times R$ und somit Paare $(p, r) \in PA$ mit $p \in P$ und $r \in R$ (siehe Kapitel 2.2.4).

Definition 1 (Rollenrechte und Rollenrechtemengen) *Seien die "Rollenrechte" P_r für $r \in R$ definiert als:*

$$P_r =_{def} \{p \mid (p, r) \in PA\}$$

und die Menge alle "Rollenrechtemengen" mit $X \subseteq R$:

$$P_X =_{def} \bigcup_{r \in X} P_r$$

Über geeignete Algorithmen soll nun aus der Menge P_r eine den neu zu definierenden Rollenrechten C (Candidates) ähnliche Menge gefunden werden, um zu verhindern, dass P_r anwächst und ungewollte Redundanzen enthält.

Beispiele für die folgenden Betrachtungen

Wir nehmen folgende Rollenrechte aus den bereits definierten Rollen an:

$$\begin{aligned} P_1 &= \{read, write, edit, delete\} \\ P_2 &= \{read\} \\ P_3 &= \{read, write\} \\ P_4 &= \{delete\} \end{aligned}$$

Betrachtet werden ferner Rollenrechte, die wie folgt neu definiert werden sollen:

$$\begin{aligned} C_1 &= \{read\} \\ C_2 &= \{read, write, edit\} \\ C_3 &= \{read, delete\} \end{aligned}$$

Ähnlichkeiten von Rollen

Große Rollenmodelle können mehrere tausend oder zehntausend Rollen beinhalten, die wiederum hunderte oder tausende von Rechten beinhalten [24]. Allerdings handelt es sich hierbei um keine Größenordnungen, die nicht effizient über einen iterativen Algorithmus gelöst werden können, der die neu zu definierenden Rollenrechte C_n mit jedem Element aus P_r vergleicht. Dabei stellt sich jedoch die Frage, wie dieser Vergleich durchzuführen ist.

Definition 2 (Ähnlichkeit von Rollenrechten) *Zwei Rollen A, B sind ähnlich ($S(A, B)$, similarity), wenn die Schnittmenge der an die Rollen assoziierten Rechte groß ist. Je größer die Schnittmenge ist, desto ähnlicher sind sich die Rollen.*

$$S(R_1, R_2) > S(R_1, R_2') \Leftrightarrow |P_{(R_1)} \cap P_{(R_2)}| > |P_{(R_1)} \cap P_{(R_2')}| \tag{4.1}$$

Die Rollenrechte sind gleich, wenn die Menge der assoziierten Rechte gleich ist.

$$S(R_1, R_2) = S(R_1, R_2') \Leftrightarrow |P_{(R_1)} \cap P_{(R_2)}| = |P_{(R_1)} \cap P_{(R_2')}| \tag{4.2}$$

Diese Definition lässt die Menge der Rollenmitglieder sowie die Bezeichnung der Rolle außer Acht, da diese Attribute für die Reduzierung von Redundanzen beim xRE Vorgehen nicht relevant sind.

Will man die Redundanz von Rollen in Bezug auf ihre Mitglieder bestimmen, greifen dieselben hier vorgestellten Ansätze. Da xRE jedoch auf der Basis von zuvor ausgewählten Testpersonen arbeitet und keine vollständige Mitgliederliste erwartet, kann an dieser Stelle der Rollenmodellierung noch nicht auf Mitgliederredundanz geprüft werden. Eine solche Prüfung kann später bei Rollenzuteilung stattfinden.

Untersucht man die Ähnlichkeit in Rollenhierachien, so muss zunächst die Vereinigung der Rechtemengen aller Rollen gebildet werden, in denen die betrachtete Rolle enthalten ist (siehe Abschnitt 4.6). Die Behandlung von Nebenbedingungen wird in Abschnitt 4.5 diskutiert.

Betrachten wir das oben gewählte Beispiel, so ist P_2 gleich C_1, P_2 sehr verschieden von P_4 aber C_2 ähnlich P_1 oder P_3.

Um die Strukturverwalter nicht durch die Software dazu zu verleiten, Personen zu viele Rechte zu geben, soll das System vorrangig solche Rollen vorschlagen, die weniger Rechte enthalten. So soll der Rollenfindungsdruide für C_2 eher P_3 vorschlagen als P_1. Stimmt der Rollenverwalter dem Druiden nämlich zu, dass P_3 statt C_2 benutzt werden kann, um nicht neue Rollen definieren zu müssen, erhalten die neuen Mitglieder auch das Recht *delete*, das im Anwendungsfall des Rollenverwalters eventuell gar nicht betrachtet wurde.

Manhattan-Distanz

In [54] sind einige Ähnlichkeitsfunktionen für den Vergleich multimedialer Daten zusammengestellt. Überführt man die Rollenrechte jeweils in einen Vektor, so lässt sich die Manhattan-Distanz auf diese Vektoren anwenden.

Für die Überführung bringen wir die Menge aller möglichen Rechte in eine Reihenfolge. Im Vektor wird nun ein nicht vorhandenes Recht mit einer 0 belegt und ein gesetztes Recht mit einer 1. So ergibt sich aus dem o.g. Beispiel die in der folgenden Tabelle dargestellte Belegung:

	P_1	P_2	P_3	P_4
read	1	1	1	0
write	1	0	1	0
edit	1	0	0	0
delete	1	0	0	1

Die Manhattan- oder City block-Metrik ist wie folgt definiert:

$$d(x, y) =_{def} \sum_{i=1}^{n} |x_i - y_i| \tag{4.3}$$

Die Differenz zwischen den beiden Vektoren gibt jeweils an, in wie vielen Rechten sich die beiden Vektoren unterscheiden. Es ergibt sich folgende Ergebnistabelle:

x / y	P_1	P_2	P_3	P_4
C_1	3	0	1	2
C_2	1	2	1	4
C_3	2	1	2	1

Die aus dieser Matrix abgeleiteten Vorschläge für den Rollenadministrator sind leicht nachzuvollziehen: Statt $C_1 = \{read\}$ wird $P_2 = \{read\}$ angeboten. Die Differenz zwischen den beiden Rollen beträgt 0. Statt $C_2 = \{read, write, edit\}$ neu zu definieren, kann $P_1 = \{read, write, edit, delete\}$ oder $P_3 = \{read, write\}$ genutzt werden. Beide haben eine Differenz von 1 zum Kandidaten. Bei P_1 würde den Benutzern zusätzlich das Recht *delete* angeboten werden, bei P_3 müssten sie ohne *edit* auskommen[1].

Um nun zu erreichen, dass vornehmlich solche Rollen als ähnlich klassifiziert werden, die weniger Rechte besitzen, können beim Vergleich Gewichte hinzugefügt werden. Zu beachten ist dann jedoch, dass damit die Kommutativität nicht mehr gegeben ist.

[1] Sinnvoll könnte z.B. auch eine Teilung von C_2 in P_3 und eine neue Rolle $C_4 = \{edit\}$ sein. Um die Beispiele einfach zu halten, werden wir solche Optimierungen jedoch hier nicht betrachten. Die Thematik wird auf Seite 84 betrachtet.

Definition 3 (Modifizierte Manhattan-Metrik) *Wir modifizieren die Manhattan-Metrik wie folgt:*

$$d'(x,y) =_{def} \sum_{i=1}^{n} |x_i - \omega y_i| \tag{4.4}$$

Die neuen Rollen sind nun jeweils als x einzusetzen, wogegen die bestehenden Vergleichsrollen für y eingesetzt werden. Mit $\omega = 2$, ergibt sich folgende veränderte Ergebnistabelle:

x / y	P_1	P_2	P_3	P_4
C_1	7	1	3	3
C_2	5	3	3	5
C_3	6	2	4	2

Wie wir sehen, unterscheidet sich der Vorschlag P_2 statt C_1 nicht. Für C_2 wird jetzt jedoch P_2 oder P_3 angeboten. Die Rollen mit mehr Rechten erhalten nun sehr große Differenzwerte.

Wie stark der Vorschlag einer Rolle mit mehr Rechten bestraft wird, kann mit der Festlegung von ω gesteuert werden. Die Nutzung des ω als zu definierende Konstante verschiebt das ursprüngliche Problem auf die Ermittlung eines geeigneten Wertes, so dass der Algorithmus an dieser Stelle noch verbessert werden muss (siehe "Auswahl eines geeigneten Algorithmus").

Hamming-Abstand

Im Gegensatz zur Manhatten-Distanz ist der Hamming-Abstand [101] auf Binärdaten definiert. Zwar funktioniert, wie das Beispiel oben zeigt, die Manhatten-Distanz gut für den Anwendungszweck, da die Vektoren jedoch ohnehin mit $\{0,1\}$ belegt sind, ist der Hamming-Abstand ebenso geeignet:

$$\Delta(x,y) =_{def} \sum_{x_i \neq y_i} 1$$

Im gegebenen Anwendungsfall liefern die beiden Funktionen equivalente Ergebnisse. Die Ergebnistabellen für die gegebenen Beispiele sind folglich identisch.

Mengentheoretischer Ansatz

Da in [85] das Rollenmodell über Mengen definiert ist und ferner Definition 2 die Ähnlichkeit über Mengen beschreibt, liegt eine Lösung nahe, die direkt auf Mengen basiert.

Definition 4 (Differenz von Rollenrechten über Mengen) *Der Abstand zwischen zwei Rollenrechten lässt sich wie folgt berechnen:*

$$D(X,Y) =_{def} |X \cup Y| - |X \cap Y| \tag{4.5}$$

Gewichtung der Rechte

Wie bereits erwähnt, ist die Ermittlung eines "geeigneten" ω für die modifizierte Manhattan-Metrik (Definition 3) eine Verlagerung der ursprünglichen Problemstellung. Die Lösung dieser Fragestellung besteht in einer variablen Belegung von ω. Gewichtet man die Rechte selbst und setzt die jeweilige Gewichtung für ω ein, wird nicht nur die Frage nach einem "geeigneten" ω gelöst, die Qualität der Vorschläge des Druiden können über dieses Verfahren ferner verbessert werden.

Für die Gewichtung der Rechte kann eine Klassifizierung ähnlich dem Bell-LaPadula Modell verwendet werden.

Im Jahr 1973 veröffentlichten D. E. Bell und L. J. LaPadula einen Fachartikel über mathematische Grundlagen der Computersicherheit [9]. Das später Bell-LaPadula genannte Modell geht von so genannten "clearance level" aus. Dieses im Militär übliche Konzept klassifiziert verschiedene Objekte als (z.B. "öffentlich", "geheim", "streng geheim"). Zugriff auf solche Dokumente hat nur, wer selbst genügend hoch klassifiziert ist.

Wir ordnen jedem Zugriffsrecht (Permission) eine Schutzklasse (Protection Class) zu. So können die Zugriffsrechte aus dem Beispiel wie folgt definiert werden: $(read, uncritically)$, $(write, sensitive)$, $(edit, critical)$, $(delete, critical)$. Den Klassifizierungen werden nun numerische Werte (Zweierpotenzen) zugewiesen: $uncritically = 1$, $sensitive = 2$, $critical = 4$.

Die Zuordnung der Klassifizierung kann wie in unserem Beispiel intuitiv erfolgen oder mittels quantitativer Verfahren ermittelt werden. Zur Ermittlung kann das Verfahren zur Quantifizierung von Sicherheitsvorfällen [100] eingesetzt werden, um die Sensibilität des Rechtes zu bestimmen. Für die Klassifizierung werden die ermittelten potentiellen Schäden in normalisierter Form eingesetzt.

Verwendet man die Manhatten-Distanz (Gleichung 4.3) und trägt statt einer "1" für die Existenz eines Rechtes in den Vektor, den dem Recht zugeordneten Klassifizierungswert ein, so ist die Tabelle nun wie folgt belegt,

	P_1	P_2	P_3	P_4
read	1	1	1	0
write	2	0	2	0
edit	4	0	0	0
delete	4	0	0	4

damit ändern sich die Abstände zu den Rollenkandidaten wie folgt:

x / y	P_1	P_2	P_3	P_4
C_1	10	0	2	5
C_2	4	6	4	11
C_3	6	4	6	1

Dieses Modell sorgt dafür, dass Rollen mit kritischen Rechten einen größeren Abstand bekommen. Das gilt hierbei allerdings sowohl für das Hinzufügen wie auch für das Weglassen von Rechten. Wie man aus diesen Ergebnissen ablesen kann, ist die Güte der Vorschläge nicht entscheidend besser, als bei den zuvor vorgestellten Verfahren.

Der Vorschlag von P_2 an Stelle von C_1 wird auch bei diesem Ansatz geliefert. Für $C_3 = \{read, delete\}$ wird jedoch $P_4 = \{delete\}$ angeboten, nicht aber $P_2 = \{read\}$, welches wir bei einer händischen Auswahl bevorzugen würden. Zwei kritische Rechte sind sich in diesem Modell näher, als zwei unkritische.

Das angestrebte Ziel erreicht man, indem die modifizierte Formel 4.4 verwendet wird und jeweils für ω der zugehörigen Klassifizierungswert ($\omega_i = \{1, 2, 4, 4\}$) eingesetzt wird.

Definition 5 (Modifizierte Manhattan-Metrik mit variabler Gewichtung) *Mit den Binärvektoren x und y und dem Vektor mit gewichteten Rechten $\omega_{1..n}$ wird die Rollenrechtedifferenz wie folgt definiert:*

$$d''(x, y) =_{def} \sum_{i=1}^{n} |x_i - \omega_i y_i| \tag{4.6}$$

Für P und C werden die zuvor verwendeten Binärmatrizen verwendet. Dadurch ergibt sich:

x / y	P_1	P_2	P_3	P_4
C_1	10	0	2	5
C_2	8	2	2	7
C_3	9	1	3	4

In diesem Modell entsprechen die Vorschläge den anfangs beschriebenen Erwartungen.

Kandidat aus Usecase	Vorschläge aus vorhandenen Rollen
$C_1 = \{read\}$	$P_2 = \{read\}$
$C_2 = \{read, write, edit\}$	$P_2 = \{read\}$ oder $P_3 = \{read, write\}$
$C_3 = \{read, delete\}$	$P_2 = \{read\}$

Tabelle 4.3: Vorschlagliste gemäß empfohlenem Modell

Grundsätzlich ist bei jeglicher Art der Gewichtung zu beachten, dass sich bei der Distanzberechnung eine Summe ergeben kann, die einer durch das Gewicht entstandenen Distanz entspricht (z.B. $2 \times sensitive = 1 \times critical$). Dadurch können scheinbar völlig sinnlose Ähnlichkeiten entstehen. Solche Effekte können dadurch verhindert werden, dass die Gewichte größer gewählt werden, als die mögliche Summe der jeweils kleineren Werte (z.B. bei vier Rechten Gewichte von 1, 5, 21, 85, ...).

Auswahl des geeigneten Algorithmus

Welche der genannten mathematischen Ansätze zur Implementierung genutzt werden, hängt von der Programmierumgebung und der Frage ab, ob und in welcher Form mit gewichteten Ergebnissen gearbeitet werden soll. Der im Rahmen dieser Arbeit erstellte Prototyp arbeitet mit Mengen, die in Python gut abgebildet werden können. Zur Berechnung werden die Mengen dann in Binärvektoren konvertiert und eine einfache Manhattan-Metrik angewendet. Der Abschnitt 4.7.1 erörtert die Ergebnisse, die mit dieser einfachen Variante des Algorithmus erzielt werden können.

Ist eine Gewichtung gewünscht, so sollte die Variante 4.6 genutzt werden. Dabei können die Gewichte außerhalb des Rollenmodells z.B. in einer Konfigurationsdatei oder einer gesonderten Datenbanktabelle gehalten werden. Das Modell selbst bleibt unberührt.

4.3 Werkzeuge zur Unterstützung des Vorgehensmodells

Im Folgenden wird die Arbeitsweise der Werkzeuge zur Unterstützung des Vorgehensmodells definiert, ohne dabei auf Implementierungsdetails einzugehen. Zur Einordnung der Werkzeuge in das Vorgehensmodell kann die Übersicht Abb. 4.1 und die Zuordnung aus Tabelle 4.2 herangezogen werden.

Die xRE Werkzeuge arbeiten, wie in Abb. 4.5 dargestellt, auf Teilen des RBAC-Modells. Zur Kontrolle der xRE Prozesse muss zusätzlich ein Datenmodell mit Verweisen auf das RBAC-Modell erstellt werden. Das xRE Datenmodell definiert Anwendungsfälle, mit denen Geschäftsrollen und Mitglieder verknüpft sind. Ein Anwendungsfall enthält ferner die Story, Protokolldaten und die Testmatrix für den Anwendungsfall. Anwendungsfälle können wiederum Anwendungsfälle referenzieren, wenn bei der Restrukturierung weitere Änderungen impliziert werden.

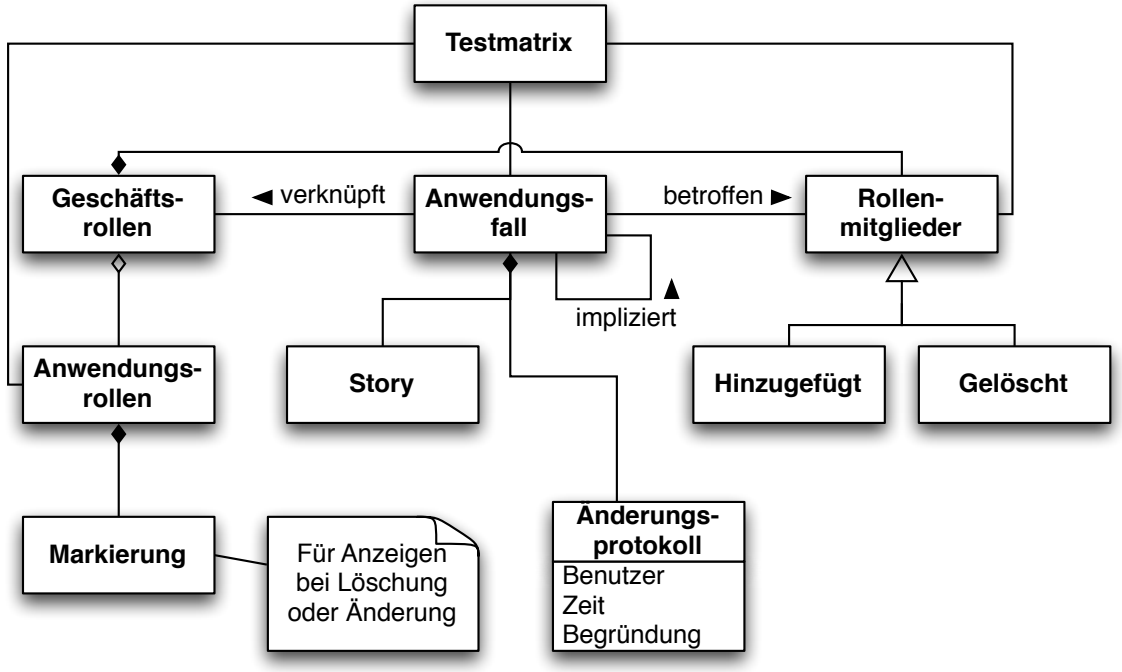

Abbildung 4.6: Klassendiagramm: xRE Datenmodell

4.3.1 xRE Verwalter

Das Modul xRE Verwalter steuert die xRE Prozesse. Es führt den Benutzer durch die einzelnen Phasen des xRE Vorgehensmodells. Der xRE-Verwalter speichert den aktuellen Zustand innerhalb eines xRE Prozesses.

4.3.2 Storyboard

Das Storyboard-Modul kann jeder Zeit von jedem Benutzer aufgerufen werden. Es dient der Erfassung von Anforderungen an das RBAC-Modell in Form von Stories (vergl. Abschnitt

"Planugsspiel" aus Anforderungskapitel 4.1.4, Seite 68). Neben dem beschreibenden Text wird die Einheit erfasst, für die das beschriebene Szenario ermöglicht werden soll. Anwendungsverwalter können ferner Stories für eine Menge von Einheiten erstellen, wenn diesen Einheiten Anwendungsrollen zugeordnet oder entzogen werden.

Der Strukturverwalter kann nun ggf. in Zusammenarbeit mit den Benutzern oder Anwendungsverwaltern eine Priorisierung der Stories durchführen und falls möglich, Stories zusammenfassen. Entsprechend der Priorisierung wird dann eine Rollenmodellierungsiteration nach der anderen durchgeführt.

4.3.3 Rollenfindungsdruide

Der Rollenfindungsdruide (kurz: Rollendruide) soll den Strukturverwalter dabei unterstützen, die Anzahl redundanter Rollendefinitionen zu minimieren. Im konkreten Anwendungsfall kann es sinnvoll sein, redundante Rollen zu definieren, weil es beispielsweise geplant ist, jeweils andere Personen zu Mitgliedern in diesen Rollen zu machen, die sich hierüber zuordnen lassen. Man darf nicht vergessen, dass selbst der Rollenname eine Kontextinformation ist, die der Rollenadministrator für sich zur Strukturierung nutzen kann. Der Rollendruide wird daher immer nur Vorschläge unterbreiten.

Die Optimierung beinhaltet zwei Phasen:

1. Ähnliche Kandidaten finden und versuchen, diese zusammenzufassen. Wurden z.B. für verschiedene Benutzer die selben Rechte in der Matrix gewählt, wird aus diesen Kandidaten ein einziger Kandidat gebildet und die jeweiligen Benutzer zu Mitgliedern dieses Kandidaten gemacht.

2. Kandidatenrollen mit den bestehenden Rollen vergleichen und ähnliche Rollen finden. Eine Kandidatenrolle kann mit der bestehenden Rolle dann auf verschiedene Arten kombiniert werden.

Für Phase 1 muss ein Schwellwert empirisch ermittelt werden, bei welcher Gleichheit der Rollendefinitionen eine Zusammenlegung vorgeschlagen wird. Mindestens kann hier jedoch vorgeschlagen werden, Rollen mit vollständiger Übereinstimmung zusammenzufassen.

Phase 2 kann mit dem in Abbildung 4.7 skizzierten Algorithmus implementiert werden. Hierbei haben die verwendeten Variablen folgende Bedeutung:

`cands` Vektorenliste mit allen Kandidaten aus den definierten Testfällen.

`roles` Vektorenliste mit allen bereits definierten Rollen, die sich im System befinden.

`c` Laufvariable, die den aktuell untersuchten Kandidaten enthält.

`r` Laufvariable, die die aktuell untersuchte Vergleichsrolle enthält.

`min` Der bislang geringste Abstand zwischen `c` und `r`.

`sim_list` Liste mit den ähnlichsten Rollen (Rollen mit Abstand `min`)

Der Algorithmus durchläuft alle Kandidaten und sucht für jeden Kandidaten aus den existierenden Rollen diejenige, die den geringsten Abstand zum Kandidaten besitzt. Dann wird dem Benutzer vorgeschlagen, anstelle des Kandidaten die gefundene Rolle zu verwenden.

Zur Vermeidung von ähnlichen Rollen ist eine Refaktorisierung des Teilmodells nötig. Dabei stehen folgende Refaktorisierungsvarianten zur Verfügung:

Rollen zusammenfassen: Das Zusammenfassen von Rollen ist der einfachste Fall zur Redundanzenvermeidung. C soll die Rechte $P_1, P_2, ..., P_n$ enthalten. Der Druide findet eine Rolle R, die $P_1, P_2, .., P_{n-x}$ Rechte beinhaltet. Die Rolle R wird um P_n erweitert. Dabei ist zu prüfen, ob alle Mitglieder von R das Recht P_n besitzen dürfen. Ist dies nicht der Fall, ist zu prüfen, ob R und C kombiniert werden können (siehe "Rollen kombinieren" weiter unten).

Rollen teilen: Die neue Kandidatenrolle C soll die Rechte $P_1, P_2, ..., P_n$ enthalten. Der Druide findet eine Rolle R, die jedoch $P_1, P_2, ..., P_{n+x}$ Rechte beinhaltet. Dieser Fall kann durch Teilen von R in R_1 und R_2 gelöst werden, wobei R_1 die Rechte $P_1, ..., P_n$ und R_2 die Rechte $P_{n+1}, P_{n+2}, ..., P_{n+x}$ zugewiesen werden. Die Mitglieder von R müssen dabei für beide Teilrollen übernommen werden. Für C und R_1 muss die Mitgliedermenge in R_1 vereinigt werden.

Würde man C ohne jede Refaktorisierung als neue Rolle hinzufügen, hätte dies ebenfalls zwei neue Rollen im Modell zur Folge. Auf den ersten Blick ist hier keine Verbesserung durch die Refaktorisierung zu erkennen. Die Teilung der Rollen zahlt sich jedoch beim Bearbeiten weiterer Szenarien aus. Die beiden geteilten Rollen sind sich nicht mehr ähnlich. Es entstehen durch die Refaktorisierung also zwei nicht redundante Rollen, die jeweils in folgenden xRE Iterationen ausgebaut werden können.

Rollen kombinieren: C soll die Rechte $P_1, ..., P_n$ enthalten. Der Druide findet eine Rolle R, die $P_1, ..., P_{n-x}$ Rechte beinhaltet. Nicht alle Mitglieder von R dürfen P_n erhalten. R bleibt unverändert bestehen und wird um die Mitglieder von C erweitert. Ferner wird eine Rolle R' erzeugt, die P_n und alle Mitglieder aus C enthält.

Wie beim Teilen entstehen hierbei zwei Rollen, die sich jedoch nicht ähnlich sind. Eine Redundanz wurde an dieser Stelle vermieden.

4.3.4 xRE UNIT

Das xRE UNIT Modul wird im xRE Prozess für zwei Funktionen benötigt:

1. Definition der Testmatrix für den Anwendungsfall.
2. Test der Rollendefinitionen nach Verfeinerung gegen die Testmatrix.

Aus der Menge aller Rechte P wählt der Strukturverwalter die für den Testfall relevanten Rechte $P' \subseteq P$ aus. Hierbei handelt es sich typischer Weise um Rechte einer bestimmten Anwendung oder um jüngst hinzugefügte Rechte. Ferner werden Testpersonen U' (Users) ausgewählt, die mit Rechten aus P' ausgestattet werden sollen. Die Testpersonen stammen typischer Weise aus der Organisationseinheit des Strukturverwalters. Die Zugehörigkeit ist jedoch nicht zwingend. xRE UNIT erzeugt nun eine Matrix $T_{(u,p)}$ mit $u \in U'$ und $p \in P'$. Für jedes $T_{(x,y)}$ wird mittels Selektion auf dem Bildschirm ein Wahrheitswert definiert.

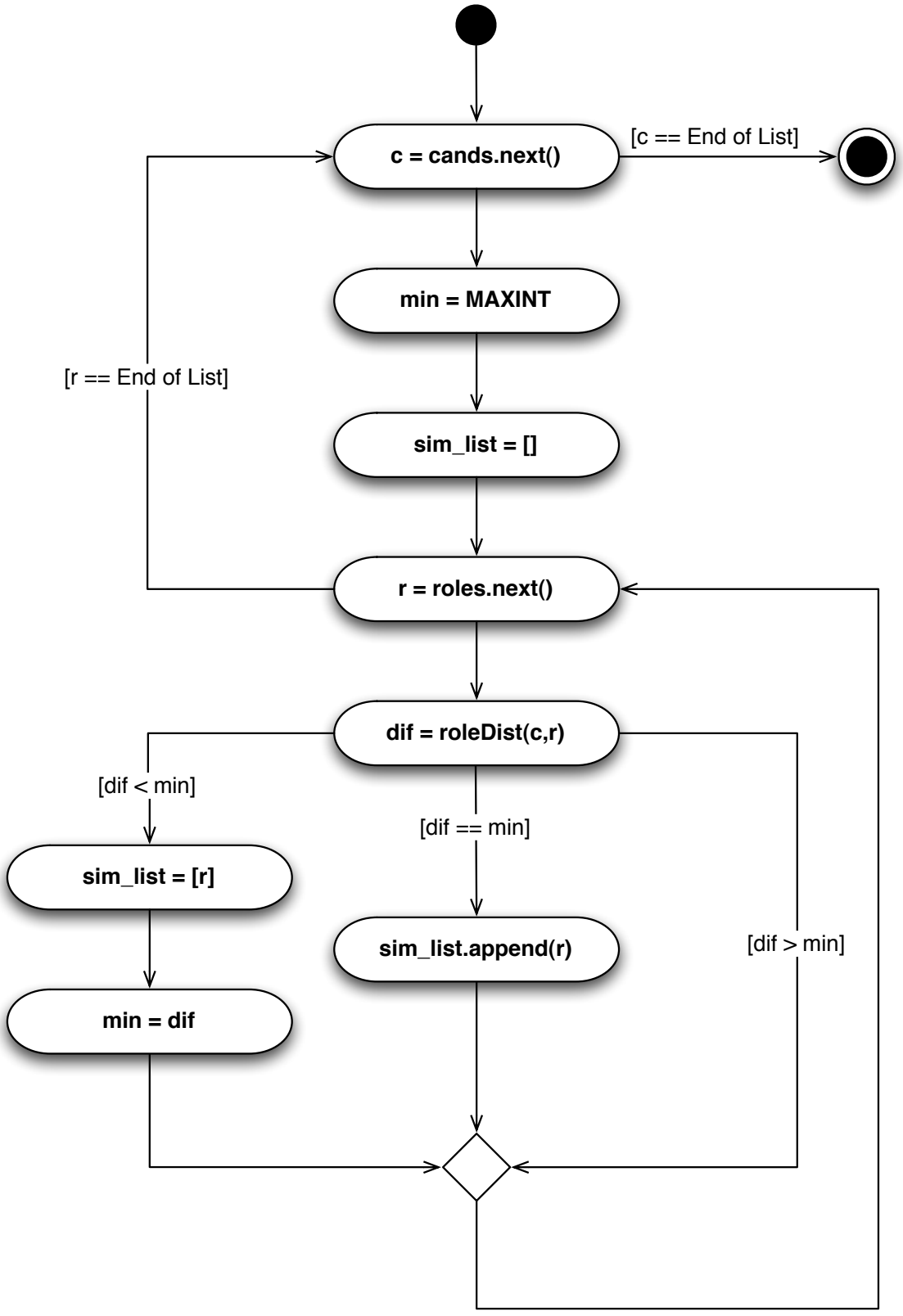

Abbildung 4.7: Algorithmus für den Rollenvorschlag

Aus der so entstandenen Matrix extrahiert der Rollenfindungsdruide im Folgeschritt die Rollenkandidaten C_x durch Zusammenfassung gleicher bzw. ähnlicher Belegungen aus P' (Kombinationen von Rechten) für verschiedene Benutzer U'. Der Rollenkandidat fasst dann die Benutzer mit gleicher P'-Belegung als Mitglieder zusammen.

Am Ende der Verfeinerung des Rollenmodells wird die Belegung aus der Matrix gegen das Modell geprüft. In der Verfeinerungsphase kann mittlerweile ein Kandidat C als Rolle aufgenommen oder durch Refaktorisierung mit in das Modell eingegangen sein. Das verfeinerte Modell muss positiv gegen die Matrix T getestet werden können. Abweichungen vom Test werden angezeigt. Es bestehen dann folgende Möglichkeiten:

Rücksetzen in den Zustand vor der Verfeinerung: Bei der Verfeinerung ist ein Fehler unterlaufen, der nicht über eine der folgenden Aktionen korrigiert werden kann. Die Verfeinerung muss erneut vorgenommen werden.

Korrektur der Testmatrix: Durch die Vorschläge des Rollenfindungsdruiden stellte sich heraus, dass die Belegung der Testmatrix fehlerhaft war. Bei den angezeigten Fehlern handelt es sich folglich um "false positives".

Korrektur der Rollendefinitionen: Die angezeigten Fehler lassen sich durch Rückkehr in den Verfeinerungsprozess beheben. Es können Korrekturen in der Rechte- und Mitgliederdefinition vorgenommen werden.

Nach erfolgreicher Prüfung des Modells gegen die Testmatrix des aktuellen Anwendungsfalls, wird das Modell gegen alle Matrixen der vorherigen Anwendungsfälle getestet, um zu prüfen, ob durch die Refaktorisierung Inkonsistenzen in den anderen Anwendungsfällen entstanden sind. Im Fehlerfall stehen jeweils erneut die o.g. Möglichkeiten zur Konfliktlösung zur Verfügung.

4.3.5 Sandbox

Während xRE UNIT jeweils nur die in den Testmatrixen enthaltenen Benutzer und Rechte betrachtet, wird in der Sandbox das gesamte Modell bzw. Teilmodell der Organisationseinheit zur Verfügung gestellt. In der Sandbox hat der Strukturverwalter folgende Möglichkeiten:

- Exploration des Modells mit allen Geschäftsrollen der Organisationseinheit, den Rollenmitgliedern und den assoziierten Rechten.
- Anzeige aller Rechte, die ein Benutzer besitzt.
- Anzeige aller Benutzer, die ein bestimmtes Recht besitzen.
- Simulation von Zugriffen ausgewählter Benutzer auf Anwendungen.

Die Sandbox verhält sich in weiten Teilen wie das Produktivsystem. Sie dient der Absicherung der strukturierten Tests von xRE UNIT. Die Exploration des überarbeiteten Modells soll verhindern, dass durch fehlerhafte Definition von Testfällen daraus abgeleitete Refaktorisierungen entstehen, die dann erfolgreich gegen die fehlerhafte Testmatrix getestet werden.

4.3.6 xRE Refaktorisierungswerkzeuge

Bei den xRE Refaktorisierungswerkzeugen handelt es sich um Analysewerkzeuge, die beispielsweise in der Sandbox zur Anwendung kommen, die aber auch unabhängig eingesetzt werden können, um den Zustand des RBAC-Modells zu untersuchen.

Trace-Tracker

Der Trace-Tracker ermöglicht die Exploration des Rollenmodells durch Folgen von Referenzen im Modell. Das Modell wird jeweils aus Sicht eines Modellobjektes dargestellt. Es besteht jeweils die Möglichkeit, vom aktuellen Objekt zu allen referenzierten oder rückreferenzierten Objekten zu wechseln. Es wird also jeweils visualisiert, auf welche Objekte vom aktuellen Objekt aus verwiesen wird und welche Objekte auf das aktuelle Objekt verweisen.

Der Trace-Tracker stellt eine Art der Visualisierung des RBAC-Modells zur Verfügung, die als virtueller Weg durch das Modell beschrieben werden kann.

User-Permission-Tester

Das User-Permission-Tester Modul bietet die Möglichkeit, die resultierenden Personen-Rechte Verknüpfungen zu überprüfen, die durch die Rollendefinitionen zustande kommen. Dabei kennt das Modul zwei Modi. Im einen Modus kann ein Benutzer ausgewählt werden. Das Modul vereinigt dann die Menge aller Rechte, die mit den Rollen assoziiert sind, in denen der Benutzer direkt oder indirekt Mitglied ist. So kann eine Rollendefinition stichprobenartig überprüft werden. Dieser Modus ist auch für Supportanfragen durch einen Benutzer interessant.

Im zweiten Modus kann aus der Menge aller Rechte ein Recht ausgewählt werden, zu dem schließlich alle Benutzer angezeigt werden, die über irgendeine Rollenmitgliedschaft das selektierte Recht besitzen. Auch dieser Modus dient der stichprobenartigen Überprüfung des Modells oder kann zur Fehlerfindung genutzt werden.

Rollendruide

Im Rollenfindungsdruiden werden an verschiedenen Stellen ähnliche Rollen gesucht. Der Rollenfinder stellt diesen Teil unabhängig vom Prozess zur Verfügung. Nach einer Definition eines Rechtevektors werden Rollen angezeigt, die diesem Rechtevektor ähnlich sind. Ferner kann eine beliebige Rolle ausgewählt werden, zu der ähnliche Rollen ermittelt werden.

4.4 Effizienz der Algorithmen

Die Frage des Einsatzes von Werkzeugen in verschiedenen Umfeldern ist auch von der Effizienz der verfügbaren Algorithmen abhängig. Der Aufwand begrenzt die Einsetzbarkeit der Software. Im Folgenden werden die Algorithmen des im Rahmen der Arbeit erstellten Prototypen zugrunde gelegt, um den Aufwand für xRE Werkzeuge abzuschätzen.

Für die Beurteilung des zeitlichen Aufwands eines Algorithmus ist eine Abschätzung mittels asymptotischer Komplexität und die so genannte O-Notation üblich. Nachfolgend werden die relevanten Routinen des xRE Controllers und der xRE Benutzerschnittstelle skizziert und eine Abschätzung des Aufwands daraus abgeleitet.

Die Aufwandsabschätzung ist im Pseudocode jeweils in geschweiften Klammern hinter den jeweiligen Zeilen angegeben.

rollendistanz()

Berechnet die Distanz zweier Rollen an Hand der übergebenen Rechtemengen $setA$ und $setB$.

```
referenceSet ← setA ∪ setB {n ← len(referenceSet)}
for r in referenceSet do {O(n)}
   x.add(1) wenn r ∈ setA, x.add(0) sonst
   y.add(1) wenn r ∈ setB, y.add(0) sonst
end for
return  vektordistanz(x, y) {O(n) Manhattan Metrik}
```

n: Anzahl der Rechte, die zu vergleichen sind.

$$O_{rollenDistanz} = O(max(O(n), O(n))) \Rightarrow O(n)$$

vereinigeGleicheKandidaten()

Fasst Rollenkandidaten aus der Testmatrix zusammen.

```
while (candidateStack ≠ ∅) do {O(o)}
   x ← candidateStack.pop()
   for y ∈ candidateStack do {O(o)}
      dif ← rollenDistanz(x.permissions, y.permissions) {O(n)}
      ... {Zusammenfassen der Kandidaten}
   end for
end while
```

o: Anzahl der Kandidaten, n: Anzahl der Rechte der Rollenkandidaten.

$$O_{vereinigeGleicheKandidaten} = O(o^2) \times O(n) \Rightarrow O(no^2)$$

verfeinerung()

Verfeinerung des Rollenmodells durch Vergleich aller Rollenkandidaten mit allen existierenden Rollen.

```
for c ∈ candidates do {O(o)}
   min ← ∞
   simList ← ∅
   for r ∈ roles do {O(p) – Rollen mit kleiner Differenz suchen}
      dif ← rollenDistanz(r.permissions, c.permissions) {O(n)}
      if dif == min then
         simList.append(r)
      end if
```

```
      if dif < min then
        simList ← ∅.append(r)
        min ← dif
      end if
    end for
    for r ∈ simList do {O(p)}
      ... {Zusammenfassen, Kombinieren, Erstellen, ...}
    end for
  end for
```

o: Anzahl der Kandidaten, n: Anzahl der Rechte in den Rollen und Kandidaten, p: Anzahl der Rollen, die bereits im Modell sind.

$$O_{verfeinerung} = O(o) \times (max((O(p) \times O(n)), O(p))) \Rightarrow O(opn)$$

xREprozess()

Aufwand des kompletten Prozesses, wie im Prototypen implementiert.

```
begrüßung() {O(1)}
modellLaden() {O(p)}
storyDefinieren() {O(1)}
testmatrixErstellen() {O_kandidatenErstellen + O_vereinigeGleicheKandidaten = O(2no²)}
verfeinerung() {O_verfeinerung = O(opn)}
sandboxTest() {O(1)}
modellSpeichern() {O(p)}
```

o: Anzahl der Kandidaten, n: Anzahl der Rechte in den Rollen und Kandidaten, p: Anzahl der Rollen, die bereits im Modell sind.

$$O_{xREprozess} = O(max(O(no^2), O(opn))) \Rightarrow O(opn) \text{ für } p \geq o$$

Ohne weitere Optimierungen ist der Aufwand im xRE-Prozess linear (vorausgesetzt es existieren mehr Rollen als Kandidaten). Die Anzahl der zu vergleichenden Rechte und Kandidaten stellen dabei in der Regel keine signifikante Größe dar. In den Tests mit produktiven Daten aus dem Rollensystem der TU Berlin waren ca. 900 Applikationsrollen (entspricht hier n) in ca. 3000 Geschäftsrollen (entspricht hier p) zu finden. Entwickelt wurde xRE für die Definition von Rollen innerhalb einer Organisationseinheit. An der TU Berlin haben Organisationseinheiten heute maximal 20 Rollen definiert (Durchschnitt 5,59 Rollen). Von den 900 Applikationsrollen werden weniger als 100 pro Einheit benutzt. Die angeführten Abschätzungen zeigen jedoch deutlich, dass selbst Einheiten mit 3000 Rollen problemlos über die xRE-Werkzeuge verwaltet werden könnten. Ein Test, der jede der 3000 Rollen im Modell mit jeder anderen Rolle verglichen hat zeigte, dass das in Python geschriebene Testprogramm auf einem Mac mini[2] nur wenige Minuten zur Erstellung einer Liste mit den Rollenabständen benötigte.

Der Speicheraufwand der Algorithmen ist ebenfalls linear und hängt allein von der Anzahl der Rollen-, Rechte- und Kandidatenobjekte ab. Der benötigte Speicherplatz für Vergleichsoperationen ist konstant und kann vernachlässigt werden.

[2] Apple Mac mini, Prozessor: 1.83 GHz Intel Core 2 Duo, 2GB RAM.

4.5 Behandlung von Nebenbedingungen

Wie in Kapitel 2.2.4 dargestellt, gehören Nebenbedingungen zu den grundlegenden Eigenschaften von RBAC-Modellen. Über Nebenbedingungen können beliebige Aussagen getroffen werden. In [24] werden Ansätze, wie z.B. zeitliche Nebenbedingungen vorgestellt. Ferner gibt es Definitionen bezüglich der Anwendbarkeit von Nebenbedingungen innerhalb eines RBAC-Modells (z.B. Bedingungen für die Mitgliedschaften in Rollen, Bedingungen für die Nutzung von Rechten innerhalb einer Rolle usw.). Die konkrete Anwendung im System unterliegt den Entwurfsentscheidungen der Systementwickler. Vor diesem Hintergrund wird die Behandlung von Nebenbedingungen im xRE rein konzeptionell betrachtet.

Wie bereits im Abschnitt "Tests" (Seite 70) angedeutet, besteht der Lösungsansatz für die Behandlung von Nebenbedingungen in xRE darin, in die Testmatritzen nicht mehr nur die Werte "0" und "1" einzusetzen, sondern an deren Stelle Funktionen für Nebenbedingungen.

Für die Anwendbarkeit der Nebenbedingungen müssen folgende Voraussetzungen erfüllt sein:

- Die Nebenbedingungen müssen durch Funktionen mit boolschen Ergebnissen ausgedrückt werden können ($f : \alpha \rightarrow \texttt{bool}$).
- Unabhängig davon, wo im Modell die Nebenbedingungen verankert sind, müssen eindeutige Rollen–Rechte- bzw. Benutzer–Rechte-Zuordnungen möglich sein, um die Funktionen in die Matrixen einsetzen zu können.
- Der Algorithmus zum Finden ähnlicher Rollen muss angepasst werden, ebenso wie die Auswertung der Testmatrix und die Darstellung für den Benutzer.

Es können zwei Arten von Nebenbedingungen unterschieden werden:

statische Nebenbedingungen: Die Wahrheitswerte der zu Grunde liegenden Funktionen können unabhängig von zur Laufzeit zu ermittelnden Parametern ausgewertet werden; d.h. die Ergebnisse lassen sich während der Rollenmodellierung bereits berechnen.

dynamische Nebenbedingungen: Die Wahrheitswerte können erst zur Laufzeit ermittelt werden, stehen also während der Rollenmodellierung nur als Variablen zur Verfügung.

Im Fall von statischen Nebenbedingungen kann nach Auswertung der Nebenbedingungsfunktionen analog zum bereits entworfenen Verfahren (gemäß 4.3) vorgegangen werden. Den einzelnen Schritten geht jeweils eine Berechnung der statischen Nebenbedingungen voran.

Für dynamische Nebenbedingungen sind folgende Modifikationen notwendig:

- An Stelle der Funktionen werden Variablen eingesetzt, die die Werte $\{0, 1\}$ annehmen können (sowohl für Rollen R_x als auch für Kandidaten C_y).
- Die Berechnung der Abstände wird mit jeder möglichen Belegung der Variablen durchgeführt.
- Die Minima, Maxima und Durchschnittswerte der Abstände werden mit den dazugehörigen Belegungen gespeichert.

- Für die Unit-Tests werden ebenfalls alle möglichen Variablenbelegungen getestet. Ein Test ist erfolgreich, wenn mindestens eine gültige Variablenbelegung gefunden wird, die die Vorgaben in der Matrix erfüllt.

Bei diesen Modifikationen bleibt unberücksichtigt, wie sich die Nebenbedingungen zur Laufzeit ggf. auch untereinander verhalten. So ist es beispielsweise denkbar, dass sich bestimmte Variablenbelegungen gegenseitig ausschließen. Abhängig von der Implementierung der Nebenbedingungen kann die Behandlung im Rahmen des xRE-Verfahrens weiter optimiert werden.

Neben den Berechnungen sind auch die Darstellungen innerhalb des xRE-Prozesses anzupassen. Bei statischen Nebenbedingungen genügt es, die Ergebnisse der Auswertung dieser Nebenbedingungen zu präsentieren. Bei dynamischen Nebenbedingungen müssen jeweils die Bedingungen angezeigt werden, unter denen das präsentierte Ergebnis gültig ist, z.B. C_1 kann mit R_5 zusammengefasst werden, sofern für C_1 die Nebenbedingung $f(x, y, z) = \texttt{true}$ ist.

Das Beispiel zeigt deutlich, dass die Behandlung der Nebenbedingungen nicht nur weitere Herausforderungen für die Implementierung der xRE-Werkzeuge bedeutet, sondern auch hohe Ansprüche an die Benutzer stellt. Für den Benutzer wäre es hilfreich, wenn bei der Definition der Nebenbedingungen für beide Wahrheitswerte beschreibende Texte hinterlegt würden (z.B. `true`: "Kein Mitglied der Rolle ist gleichzeitig Mitglied in der Rolle 'Finanzprüfung'.", `false`: "Mindestens ein Mitglied der Rolle ist gleichzeitig Mitglied in der Rolle 'Finanzprüfung'.").

Das xRE-Verfahren schließt die Behandlung von Nebenbedingungen nicht aus. Die Behandlung von statischen Nebenbedingungen fügt dem Verfahren lediglich den Schritt "Berechnung der Nebenbedingungen" hinzu. Die Modifikationen für die Behandlung dynamischer Nebenbedingungen sind hingegen weitreichender. Diese müssen ferner an die Implementierung im Rollensystem und die Anwendungsfälle angepasst werden. Vor allem die benutzerorientierte Präsentation der Ergebnisse muss dem jeweiligen Umfeld angepasst werden.

4.6 Arbeiten mit hierarchischen Rollenstrukturen

Gegen den Einsatz von hierarchischen Rollenstrukturen innerhalb der Organisationseinheiten spricht die Erhöhung der Komplexität sowohl der verwendeten Werkzeuge zur Rollenmodellierung, als auch die steigende Komplexität für den Rollenverwalter. Hierarchien widersprechen somit dem ursprünglichen Designziel von xRE, ein einfaches Verfahren für kleine Untereinheiten zur Verfügung zu stellen. Hierarchien werden jedoch durch xRE nicht ausgeschlossen. Die Verwendung von Rollenhierarchien kann die Rollenzahlen durch Erhöhung der Übersichtlichkeit reduzieren. Hierarchien eignen sich zur Abbildung von Abhängigkeiten. Ferner ist zu bedenken, dass es Anwendungsgebiete gibt, die sehr stark hierarchisch organisiert sind.

Um xRE auf Rollenhierarchien anwenden zu können, muss sowohl der Algorithmus zum Finden von ähnlichen Rollen angepasst werden, als auch die Methoden zur Refaktorisierung des Modells. In hierarchischen Rollenmodellen besitzen Rollen neben den für sie definierten Rechten auch geerbte Rechte aus den tiefer liegenden Rollenhierarchien. Bei der Refaktorisierung muss berücksichtigt werden, dass Änderungen in der Rollendefinition auch Auswirkungen auf die Kind-Rollen haben. Bei der Refaktorisierung muss nicht nur die Frage beantwortet werden, wie die Änderungen einzupflegen sind (teilen, zusammenfügen, kombinieren), sondern auch auf welcher Hierarchieebene.

Die Anpassungen der Werkzeuge lässt sich wie folgt zusammenfassen:

- Der Algorithmus zum Finden von ähnlichen Rollen muss in Bezug auf die Zusammenstellung der Rechtevektoren angepasst werden. Für jede Rolle ist die Vereinigungsmenge aller Rechte bis zur Wurzel des Baumes zu bilden; d.h. es müssen zunächst alle Rechte, sowie die geerbten Rechte zusammengestellt werden. Die folgenden Schritte bedürfen keiner weiteren Anpassung.

- Die Refaktorisierung muss erweitert werden, um Hierarchien korrekt zu behandeln.
  - Die beschriebenen Aktionen "Zusammenfassen", "Teilen", "Kombinieren" sowie die Basisfunktionen "Hinzufügen", "Ändern" und "Löschen" werden auch für Rollen in Hierarchien zur Verfügung gestellt.
  - Um zu ermitteln, an welcher Stelle im Baum zusammengefasst, geteilt oder kombiniert wird, wird jeweils die ähnlichste Rolle innerhalb des ausgewählten Teilbaums ermittelt.
  - Es wird so tief wie möglich im Baum geändert, um möglichst wenige Kindobjekte refaktorisieren zu müssen.
  - Alle Refaktorisierungsaktionen müssen jeweils auch auf alle Kindobjekte der geänderten Rolle angewandt werden.

Der Refaktorisierungsaufwand kann insbesondere aus Benutzersicht im Vergleich zur Refaktorisierung auf flachen Rollenmodellen wachsen. Fordert das Umfeld jedoch die Nutzung von Hierarchien, können diese durch das xRE unterstützt modelliert werden.

4.7 Evaluation des Verfahrens

Der Analyse und dem Entwurf des xRE Verfahrens folgt nun eine Bewertung, die sich zum einen auf Tests mit Daten aus einer Produktivumgebung sowie aus Vergleichen mit verwandten Verfahren stützt. Abschließend kann eine Empfehlung für den Einsatz des Verfahrens ausgesprochen werden.

4.7.1 Prototypische Implementierung und Tests mit Daten aus Produktivumgebung

Die xRE Werkzeuge wurden prototypisch implementiert. Zur Bewertung der Algorithmen und der Einsatzfähigkeit des Verfahrens werden Tests u.a. mit Daten aus dem Produktivsystem an der TU Berlin durchgeführt.

Versuchsbeschreibung

Die prototypische Implementierung umfasst ca. 1500 Zeilen Python Code. Für die Tests mit dem Prototypen wurden anonymisierte Daten aus dem TUBIS Produktivsystem verwendet. Dabei wurden ausschließlich Rollen betrachtet, die durch den Rollenverwalter administriert werden können. Die so genannten Standardrollen, die vom System automatisch an die jeweiligen Statusgruppen verteilt werden, wurden bei den Versuchen nicht beachtet.

Zum Zeitpunkt des Datenimports waren im System 2703 Rollen in 480 Organisationseinheiten definiert. Eine Einheit definiert durchschnittlichen 5,6 Rollen (minimal eine, maximal 19).

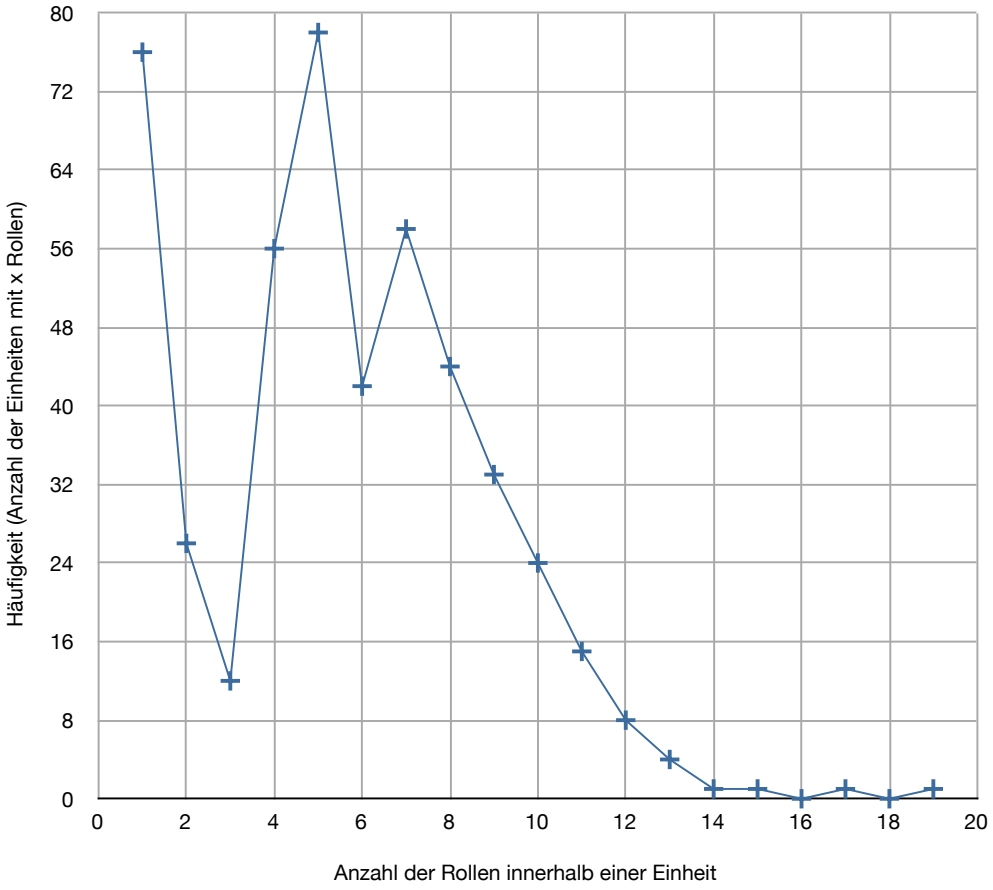

Abbildung 4.8: Verteilung der Rollenzahlen

Abb. 4.8 zeigt die Verteilung der Rollenanzahl in den Einheiten. So hatten 76 Einheiten eine Rolle definiert, 26 Einheiten zwei Rollen, 78 Einheiten hatten fünf Rollen definiert und je eine 14, 15, 17 und 19 Rollen.

Da das xRE Verfahren u.a. zum Ziel hat, Redundanzen bei der Rollendefinition zu vermeiden, wurde ferner untersucht, wie viele redundante Rollen innerhalb der Einheiten zu finden sind. Hierfür wurden die Differenzen zwischen allen Rollen innerhalb einer Einheit berechnet und diejenigen Rollen gezählt, die eine Differenz von Null aufwiesen.

Nur etwa 6% der Einheiten hatten redundante Rollen definiert. 94% der Einheiten hatten keine redundanten Rollen, 5% hatten eine redundante Rolle definiert und jeweils ca. 0,5% zwei oder drei redundante Rollen.

Die relativ geringe Zahl von redundanten Rollen lässt sich leicht aus der Verteilung der Rollenzahlen pro Einheit erklären. Mehr als die Hälfte der Einheiten besitzt 5 oder weniger Rollen. Aber auch in dieser überschaubaren Anzahl von Rollen wurden im Zuge der Evaluation redundante Rollen gefunden.

Bei steigender Rollenzahl wird auch die Redundanzvermeidung eine größere Rolle spielen. Es wird damit gerechnet, dass nach Einführung von xRE durch die einfachere und transparente Benutzerführung häufiger von der Möglichkeit Gebrauch gemacht wird, neue Rollen

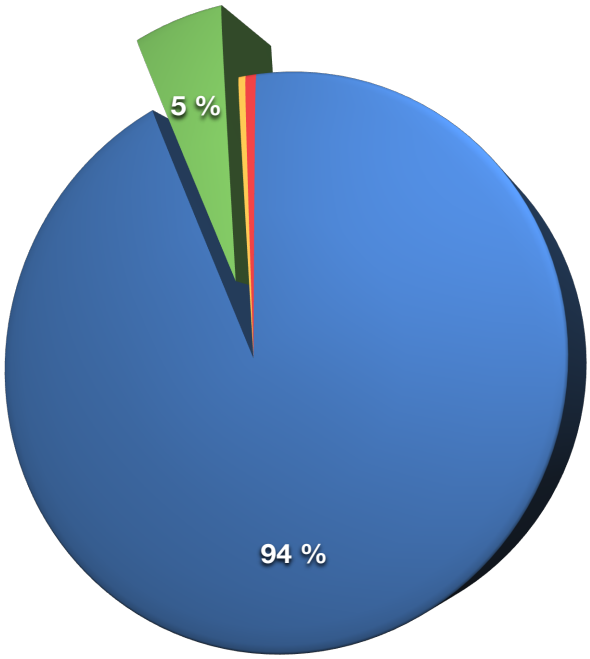

Abbildung 4.9: Rollenredundanz

zu definieren. Auch in Hinblick auf eine steigende Zahl von integrierten Anwendungen und Benutzern ist ein Verfahren willkommen, das die Zahl der Rollen klein hält und Redundanzen vermeidet.

Neben allgemeinen Tests mit dem Prototypen, bei denen die grundsätzliche Funktionsweise des Prototypen und der Algorithmen nachgewiesen wurde, sind auch Testreihen basierend auf Daten aus dem Produktivsystem durchgeführt worden. Dafür wurden drei Einheiten ausgewählt. Eine Einheit mit 5 Rollen, von denen jedoch drei redundant waren, eine Einheit mit 11 Rollen (3 redundant) und die Einheit mit 19 Rollen (ohne Redundanz).

Die Rollenkonfigurationen wurden ermittelt und danach die Rollen in den Einheiten gelöscht. Danach wurden die Konfigurationen über das xRE Verfahren in 1 - 5 Iterationsschritten wieder hinzugefügt. Danach konnte die Anzahl und die Qualität der entstandenen Rollen mit der ursprünglichen Konfiguration verglichen werden.

Diskussion der Ergebnisse

Die Tests basierend auf den Daten aus dem Produktivsystem zeigten, dass das Verfahren geeignet ist, um bei der Modellierung des Rollensystems zu unterstützen. Die aus den Ursprungsdaten abgeleiteten Anforderungen an das Rollenmodell ließen sich mit Hilfe von ein bis fünf xRE Iterationsschritten nachbilden. Dabei wiesen die mittels xRE aufgebauten Modelle keine Redundanzen mehr auf. Die Rollenzahl konnte von 5 auf 2, von 11 auf 9 und von 19 auf 17 Rollen reduziert werden. Eine händische Überprüfung zeigte jeweils nach der Modellierung, dass die Rechtekonfiguration der aus den Ausgangsdaten entsprach.

Das Verfahren lässt sich zügig durchführen. Als Rollenverwalter erlangt man schnell Routine bei der Nutzung des Verfahrens. Die gemeinsam mit dem Rollendruiden erstellten Modelle folgen jeweils der gleichen Systematik. Das erhöht die Lesbarkeit der Modelle für andere

Rollenadministratoren.

Während der Versuche wurde eine Verfeinerung in Bezug auf die Berechnung des Rollenabstands eingeführt. Der Prototyp berechnet die Rollendifferenzen faktisch nicht aus einer Matrix, wie im mathematischen Modell in 4.2.1 beschrieben, sondern optimiert den Algorithmus, indem jeweils paarweise nur die vorkommenden Rechte verglichen werden (z.B. $d((1,0,1,1),(1,1,0,0))$ statt $d((1,0,1,1,0,0,0,0),(1,1,0,0,0,0,0,0))$). Tritt eine Differenz von 2 bei einem Paar auf, so werten wir diese Differenz höher, als eine Differenz von 2 bei einem 20-Tupel. Um die Werte weiterhin vergleichbar zu halten, wird die Differenz über die Anzahl der verglichenen Rechte normiert:

$$d_{norm} = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n}$$

Fazit und Ausblick

Die prototypische Implementierung des Verfahrens und der Test mit Daten und Anforderungen aus dem Produktivsystem zeigen, dass mittels xRE unter realen Bedingungen Ergebnisse erzielt werden können, die genauso gut oder besser im Sinne der Rollenanzahl und Redundanzvermeidung sind, wie eine Modellierung, die ohne Rollenmodellierungsverfahren entstanden ist.

Die bisherigen Versuche lassen noch keinen Vergleich mit anderen Rollenmodellierungsverfahren zu. Ziel der Entwicklung von xRE war, die händische Modellierung abzulösen bzw. zu unterstützen. Die Tauglichkeit des Verfahrens hierfür ist in einigen wenigen Versuchen nachgewiesen, so dass die Versuche auf eine größere Zahl von Rollenadministratoren und Rollenmodelle im produktiven Einsatz ausgeweitet werden können.

4.7.2 Vorteile von xRE im Vergleich zu Role-Mining

Role-Mining dient der Analyse von für Anwendungen vergebenen Rechten für Benutzer mit dem Ziel, durch Clusterbildung Rollendefinitionen daraus abzuleiten. Da auch xRE zunächst Personen und Rechte abfragt, um daraus Schlussfolgerungen für die Rollendefinition zu ziehen, liegt eine Ähnlichkeit der Verfahren nahe. Tatsächlich verfolgen beide Verfahren jedoch gegensätzliche Ansätze und besitzen unterschiedliche Anwendungsgebiete.

Das Role-Mining ist ein typisches Bottom-Up-Verfahren. Es setzt die Existenz vieler existierender Benutzer–Rechte-Definitionen voraus. Daraus werden schließlich die Rollen abgeleitet. Es handelt sich also um ein deskriptives Verfahren, dass dabei unterstützt, einen gegebenen Zustand in ein Rollenmodell zu überführen.

Dem gegenüber handelt es sich beim xRE um ein Top-Down-Verfahren. Ausgangspunkt sind die Anwendungsfälle, für die Rollen definiert werden sollen. Die Benutzer-Rechte-Definitionen werden beim xRE-Verfahren jeweils nur für einen Anwendungsfall abgefragt, um daraus Definitionen ableiten oder existierende Definitionen anpassen zu können. Es handelt sich also um ein konstruktives Verfahren.

Während Role-Mining am besten mit großen Mengen an Daten funktioniert, ist xRE vor allem für kleine Gruppen und auch für Einzelfälle optimiert. Beide Verfahren sind darauf ausgerichtet, durch Software unterstützt zu werden. Auf Grund der zu verarbeitenden Datenmengen kann xRE auch leicht ohne Software eingesetzt werden, wohingegen die Verwendung von Role-Mining nur mit Softwareunterstützung sinnvoll ist.

Ein entscheidender Vorteil von xRE ist die Verknüpfung von Modell, Modelländerung und Anwendungsfällen mit der automatischen Dokumentation an Hand der Stories. Beim Role-Mining sind die entstehenden Rollen zunächst semantikfrei.

Die Stärken des Role-Mining liegen eindeutig in der Initialisierung von Rollenmodellen. Es handelt sich also um ein Werkzeug zur Migration auf ein rollenbasiertes System. Modelländerungen sind in der Folge nicht vorgesehen. Insofern kann Role-Mining z.B. bei der Initialisierung eines Modells herangezogen werden, dass später mit xRE verwaltet wird.

Auf der anderen Seite bietet eine schrittweise Migration auf Basis von xRE diverse Vorteile: Zum einen fällt beim Verfahren Dokumentation an, zum anderen können durch den konstruktiven Ansatz Rechte und ggf. auch Aufgabenverteilungen und Prozesse neu beurteilt werden, bevor sie im Rollenmodell manifestiert werden.

4.7.3 Vorteile von xRE im Vergleich zum klassischen Role-Engineering

Das klassische Role-Engineering geht davon aus, dass ein Rollenadministrator oder ein Team die Modellierung des gesamten Rollenmodells vornehmen. Ursprünglich sind im Role-Engineering keine Methoden zum Ändern des Modells vorgesehen. Unterstützt wird hier eine ganzheitliche Sichtweise, die Expertenwissen in Bezug auf das Rollenmodell ebenso wie in Bezug auf Prozesse und eingesetzte Systeme fordert. Mittlerweile sind für das Role-Engineering auch Softwarewerkzeuge zur Unterstützung bereitgestellt. Es existieren unterschiedliche Erweiterungen zur Verbesserung unterschiedlicher Aspekte beim Role-Engineering.

Die Unterstützung des verteilten Ansatzes stand bei der Entwicklung des xRE im Vordergrund. Es besteht jedoch keine Beschränkung auf Teilmodelle, so dass das Verfahren universell eingesetzt werden kann. Für die Administration von Modellteilen ist ein im Vergleich zum klassischen Role-Engineering geringes Vorwissen erforderlich. Das Verfahren setzt auf das Expertenwissen im Organisations- und Prozessumfeld. Die Unterstützung durch die Software stand dabei von Anfang an im Vordergrund, so dass das gesamte Verfahren über ein Dialogsystem genutzt werden kann.

Gegenüber dem klassischen Role-Engineering fehlt dem xRE Verfahren die ganzheitliche Sicht auf das Modell. Dafür ist es sehr viel besser automatisierbar. RBAC Experten können klassische Verfahren auch wechselnd mit den xRE Verfahren verwenden. Der Vorteil für RBAC Experten kann dabei darin bestehen, dass sie in kleinen Iterationsschritten softwaregestützt arbeiten können und dabei eine hohe Transparenz gegenüber dem Auftraggeber erreichen können. So lassen sich Rollendefinitionen gemeinsam im Team aus RBAC- und Organisationsexperten erstellen.

Den größten Nutzen bringt jedoch die Anwendung von xRE in verteilt administrierten RBAC-Umgebungen ein, insbesondere dann, wenn die Administration nicht ausschließlich von RBAC-Experten durchgeführt werden soll.

4.7.4 Allgemeine Kritikpunkte an XP und deren Bedeutung für xRE

XP treibt bestimmte Aspekte der Software Entwicklung, wie der Name schon sagt, zum Extrem. Damit löste sein Erfinder Kent Beck heftige Diskussionen aus [99]. Da xRE nach dem Vorbild von XP entstand, stellt sich die Frage, inwieweit die Kritikpunkte, die zu XP existieren, auch auf xRE zutreffen und wie dies zu bewerten ist.

Nach [99] wird für XP die Annahme getroffen, dass sich Kosten für Änderungen am zu entwickelnden System linear oder höchstens logarithmisch im Verlauf des Projektes entwi-

ckeln. Untersuchungen des US Verteidigungsministeriums aus den 60ger und 70ger Jahren gehen von exponentiellen Kostenentwicklungen aus. Aus diesem Grund sind die klassischen Entwicklungsmodelle vor allem darauf ausgerichtet, Fehler so früh wie möglich im Projekt zu vermeiden. Da Kent Beck seine Kostenannahmen nicht wissenschaftlich belegen kann, bleibt die Annahme der Kostenentwicklung zweifelhaft. Man muss davon ausgehen, dass der Skalierbarkeit von XP Projekten Grenzen gesetzt sind. Es besteht die Gefahr, dass der Refaktorisierungsaufwand bei größeren Projekten stark ansteigt. Gleiches gilt auch für xRE: Zwar wird die Refaktorisierung durch Software gestützt, jedoch besteht auch hier die Gefahr, dass der Aufwand bei großen zu ändernden Modellteilen stark ansteigt.

Ein weiterer Kritikpunkt an XP bezieht sich auf das gemeinsame Code-Eigentum. Gemäß XP kann jedes Teammitglied jeden Teil im Programm anpassen, wenn dies sinnvoll oder nötig erscheint. Es gibt keine Module, für die einzelne Personen verantwortlich sind. Dieses Paradigma erscheint vor allem dann riskant, wenn es sich um Module handelt, die wiederverwendet werden sollen. Dabei müssen oft Entwurfsentscheidungen getroffen und ggf. auch mit Hilfe von Kompromissen durchgesetzt werden, um die Kompatibilität zu anderen Systemen oder in Hinblick auf geplante Entwicklungen zu wahren. Dies sieht XP per se nicht vor. Auch bei der Modellierung von Organisationseinheiten können strategische Planungen im Konflikt zum Prinzip der einfachsten Implementierung stehen, die funktionieren kann, so wie sie von XP und auch xRE gefordert wird. Diese Schwachstelle wird durch die Größe eines typischen Teilmodells im Vergleich zur typischen Programmcodegröße relativiert. Das Problem lässt sich ferner durch die Festlegung von Grundsätzen und der gemeinsamen Diskussion der strategischen Modellierungsentscheidungen innerhalb des RBAC-Administrationsteams umgehen. Es steht jedoch außer Frage, dass dieser Kritikpunkt an XP auch für xRE gültig ist.

XP steht im drastischen Kontrast zu Entwicklungsmethoden, die z.B. einen definierten Qualitätsstandard oder formale Sicherheitsanforderungen adressieren. Während des XP Prozesses sind keine Dokumentationen geplant. Dieser Verzicht ist durch die Beobachtung begründet, dass sich in der Regel die Anforderungen innerhalb des Entwicklungsprozesses ändern und ein erheblicher Aufwand in die Nachführung der Dokumentation zu investieren ist. Im xRE Verfahren ist vorgegeben, an welchen Stellen Dokumentationen in Form von Stories zu erstellen sind. Diese werden archiviert. Falls aus Gründen der Qualitätssicherung oder aus anderen Gründen notwendig, ist es ferner möglich, umfassendere Dokumentationen in den xRE Prozess einzubetten und systematisch an jeden Änderungsschritt zu knüpfen.

Wegener schließt in [99] mit der Kritik an den Punkten, die im XP Verfahren nicht oder nicht ausreichend definiert sind. Im Unterschied hierzu ist das xRE Verfahren vollständig definiert und durch Algorithmen beschrieben. Das umfasst auch den Bereich der Refaktorisierung, der im Fall von XP insbesondere als unzulänglich beschrieben dargestellt wird.

Die zudem in [81] angeführten Kritikpunkte: "Potentielle Probleme bei der Anpassung der Tests bei Änderung der Anforderungen" und "unzureichende Dokumentation des Verfahrens selbst" treffen auf xRE nicht zu. Der Prozess der Anpassung von Tests ist in xRE definiert. Der Prozess selbst ist u.a. durch die vorliegende Arbeit dokumentiert.

Zusammenfassend ist festzuhalten, dass sowohl die offene Frage nach dem Änderungsaufwand bei größer werdenden Modellen, als auch Einschränkungen bei der Koordination von strategischen Entscheidungen durch das gemeinsame Code-Eigentum zu einer Einschränkung der Skalierbarkeit von xRE führen. Wie im Bereich der Softwareentwicklung helfen auch hier empirische Studien bei der Ermittlung der beeinflussenden Faktoren und der Grenzwerte.

4.7.5 Grenzen des xRE Verfahrens

Die Qualität der durch xRE erstellten Rollen hängt stark von der Definition der Testfälle ab, die ein Rollenadministrator definiert. Werden zu kleine Änderungen durchgeführt, das heisst in vielen Iterationsschritten immer nur einzelne Rechte hinzugefügt, neigt xRE dazu, für jedes Recht eine einzelne Rolle vorzuschlagen. Im Resultat könnten sehr viele Rollen entstehen, die dann unübersichtlich und schlecht zu warten wären. Dem entgegen steuert die Tatsache, dass xRE immer die ähnlichsten Rollen auswählt und vorschlägt, diese mit der aktuellen Rolle zusammenzufassen. Hier hängt es von den Entscheidungen des Rollenadministrators ab, ob der Fragmentierung der Rollendefinitionen entgegengewirkt wird. Für die Rollenadministratoren wäre es hilfreich, Richtlinien an die Hand zu bekommen, die auch in die Formulierungen und Vorschläge der Software mit einfließen können.

Während xRE sehr gut für die Definition von neuen Rollen oder die Erweiterung von Rollen geeignet ist, ist der Rückbau von Rollen nur begrenzt unterstützt. Durch Zusammenfassen oder Ersetzen von Rollen können Rechte entfernt und Rollenzahlen klein gehalten werden. Dem Verfahren fehlt jedoch eine gesonderte Methode zum Entfernen von Rechten oder Rollen (Anwendungsfall: "Wissenschaftliche Mitarbeiter sollen keinen Zugriff mehr auf Hardwarebestellungen erhalten."). Hierfür muss eine klassische Rollenverwaltung herangezogen werden, mit der es möglich ist, Mitglieder oder Rechte gezielt aus Rollen zu entfernen. Das xRE Verfahren kann also immer nur zusätzlich zu einem klassischen Rolleneditor eingesetzt werden. Selbstverständlich können die genannten Anwendungsfälle ebenfalls mit Dialogsystemen gesteuert werden, so dass für den Rollenadministrator eine konsistente Bedienung zur Verfügung steht. Auch könnten die xRE Werkzeuge dafür eingesetzt werden, nach solchen Änderungen das System wieder auf Ähnlichkeiten zu prüfen und ggf. weitere Optimierungen vorzuschlagen.

4.7.6 Anwendungsgebiete für xRE

Obwohl xRE für ein sehr spezielles Anwendungsgebiet entworfen wurde, ist das Verfahren auf andere Gebiete übertragbar. Vorab ist jedoch zu prüfen, ob xRE überhaupt gewünscht ist und seine Vorteile ausspielen kann und ob K.O.-Kriterien vorliegen, die eine Nutzung ausschließen.

RBAC-Systeme können gemäß unterschiedlicher Kriterien charakterisiert werden, wie z.B. Modellgröße, Art der Organisation, die modelliert wird oder wie die Verwaltung des RBAC-Systems organisiert ist. Die Modellgröße ist wiederum abhängig von der Zahl der modellierten Benutzer, der Anzahl der definierten Rollen und der jeweiligen Größe und der Anzahl der Untereinheiten, die modelliert werden. Das xRE Verfahren wurde mit Daten aus dem Produktivsystem im universitären Umfeld getestet. Es gibt keine Erkenntnisse, die harte Grenzen für die Größe des Rollenmodells definieren. Insofern kann postuliert werden, dass der Einsatz auch in größeren, wie in kleineren RBAC-Modellen möglich ist. Um eine wissenschaftlich fundierte Aussage darüber treffen zu können, müssten jedoch Untersuchungen mit unterschiedlich großen Modellen durchgeführt werden. Die Art der Organisation (Forschung, Lehre, Dienstleistung, Industrie, ...) spielt unmittelbar keine Rolle für die Anwendbarkeit von xRE. Allerdings beeinflusst die Organisationsart zum einen die unterschiedlichen Variablen der Modellgrößen, zum anderen ist die Verwaltung des Rollenmodells stark von der Organisations- bzw. Unternehmenskultur abhängig.

Die Administration von RBAC-Modellen wird häufig von einer Person oder einem Team im Auftrag durchgeführt. Neben der Tatsache, dass so sichergestellt werden kann, dass nur RBAC-Experten an der Rollendefinition arbeiten, ist es auf Grund des Arbeitsflusses möglich,

auch Richtlinien umzusetzen, die technisch nicht verankert sind. Die Rollenadministratoren können so leicht als Filter genutzt werden. Wie bereits in 4.7.3 beschrieben, schließt diese Art der Administration xRE nicht aus. Von entscheidenden Vorteilen des Verfahrens wird so jedoch kein Gebrauch gemacht. Auf der anderen Seite gibt es Organisationen, in denen durch einen stark verteilten Administrationsansatz, gekoppelt mit der Möglichkeit, in den Untereinheiten auch ohne Expertenwissen Entscheidungen über die Zugriffsrechte technisch umsetzen zu können, entscheidend zur Akzeptanz eines solchen Systems beigetragen wird.

Die folgenden Kriterien schließen die Nutzung von xRE aus:

Write-Once-Read-Many Modell: Das RBAC-Modell wird einmalig erstellt und bleibt dann abgesehen von wechselnden Rollenmitgliedschaften unverändert. In diesem Fall kann xRE zum Aufbau des Modells genutzt werden. Von den Refaktorisierungswerkzeugen kann jedoch kein Nutzen gezogen werden.

Logik der Rollenmodells hauptsächlich in Nebenbedingungen: Ist die Logik des RBAC-Modells hauptsächlich über dynamische Nebenbedingungen (siehe 4.5) abgebildet, also werden in den Matrixen häufiger Variablen, als Konstanten eingesetzt, greift der Algorithmus zum Finden von redundanten Rollen schlecht. Ohne anwendungsspezifische Anpassungen ist xRE in diesem Bereich nicht effizient einsetzbar.

Redundanz erwünscht: Es sind Anwendungsfälle denkbar, in denen die Definition von redundanten Rollen erwünscht ist. In diesem Bereich ist das Verfahren zur Vermeidung von Redundanzen aus dem xRE kontraproduktiv. Das xRE Verfahren lässt redundante Rollen zu, schlägt jedoch immer wieder vor, Redundanzen zu vermeiden.

Organisatorische oder rechtliche Gründe: Es gibt Fälle, in denen spezielle organisatorische oder rechtliche Anforderungen an die Dokumentation von Vorgängen oder Prozessen gestellt werden. Es ist vorab zu prüfen, ob diese Vorgaben dem xRE Verfahren widersprechen.

Technische Gründe: Der Einsatz der xRE Werkzeuge fordert eine Schnittstelle zum RBAC-System. Steht eine solche Schnittstelle nicht zu Verfügung, kann ersatzweise nur manuell nach xRE vorgegangen werden, sofern der Arbeitsablauf im RBAC-System dies zulässt.

Für die potentiellen Anwendungsgebiete von xRE kann zusammengefasst werden, dass zur Zeit keine Beschränkungen bezüglich der Größe des RBAC-Modells oder der Art der Organisation bekannt sind. Vor einem Einsatz sind die hier zusammengestellten K.O.-Kriterien zu prüfen. Ferner ist die Frage aufzuwerfen, ob das xRE-Verfahren dem angestrebten Arbeitsablauf entspricht und vom adressierten Benutzerkreis akzeptiert wird.

Kapitel 5

Erfahrungen mit dem umfassenden Autorisierungsmanagement

> Arroway: Because I can't. I had an experience. I can't prove it. I can't just explain it. But everything that I know as a human being, everything that I am tells me that it was real. I was given something wonderful, something that changed me forever: a vision of the universe that tells us undeniably how tiny and insignificant and how rare and precious we all are. A vision that tells us that we belong to something that is greater than ourselves, that we are not - that none of us is alone. I wish I could share that emotion, that everyone, if even for one moment, could feel that awe and humility and that hope that I felt, but... that continues to be my wish.
>
> Contact (film)

Den in den vorigen Kapiteln beschriebenen Konzepten und Verfahren liegt die praktische Umsetzung in Form eines Systems zugrunde, das seit einigen Jahren an der TU Berlin mit mehreren tausend Benutzern im Einsatz ist. Das System wurde als Nutzbarmachung der vorangegangenen EU-Projekte ins Leben gerufen. Wissenschaftliche Forschung und Entwicklung greifen hierbei beispielgebend ineinander: Das Produktivsystem beweist die praktische Nutzbarkeit der erarbeiteten Konzepte und bietet ferner eine Plattform für Entwicklungen, wie das eXtreme Role-Engineering, dessen Anforderungen erst aus dem großflächigen Einsatz des Systems hervor gegangen sind und das wiederum auf dieser Plattform erprobt werden kann. Nutzung und Entwicklung befruchten sich gegenseitig. Die vorliegende Arbeit soll u.a. den Zweck erfüllen, die wissenschaftlichen sowie praktischen Erkenntnisse allgemein zur Verfügung zu stellen.

Es werden Antworten auf die folgenden praktischen Fragen skizziert:

- Wie werden existierende Anwendungen in das System integriert?
- Wie können neue Anwendungen in Hinblick auf das umfassende Autorisierungsmanagement entworfen werden?

- Welche Designentscheidungen haben zur Auswahl der vorgestellten Technik geführt?
- Welche Erfahrungen wurden mit der Einführung des Systems gewonnen?

Aus konstruktiver Sicht gibt das folgende Kapitel einen Überblick über Technologien, die zur Umsetzung eines umfassenden Autorisierungsmanagements herangezogen werden können.
Dieses Kapitel gliedert sich gemäß des Schichtenmodells des Rahmenwerkes in Abb. 5.1:

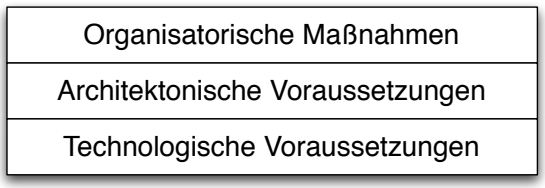

Abbildung 5.1: Schichten des Autorisierungsmanagement-Rahmenwerkes

Mit Hilfe von technologischen Mitteln und Verfahren kann eine Architektur realisiert werden, die zur Umsetzung von organisatorischen Verfahren genutzt werden kann. Das Kapitel wird mit dem Abschnitt "Der Anwendungsfall TU Berlin" geschlossen, in dem ich aus meiner Sicht bemerkenswerte Erkenntnisse aus dem Alltag der Implementierung eines Autorisierungssystems und damit auch der Schnittstelle zum Benutzer und allen Systemgrenzen zusammengestellt habe.
Die beschriebenen Methoden und Techniken sind in der Fachwelt bekannt [24, 64, 16, 82, etc.] und werden in Teilen oder in Kombination bereits vielerorts eingesetzt. Das beschriebene Rahmenwerk wurde von meiner Arbeitsgruppe in einigen internationalen Drittmittelprojekten prototypisch erprobt und im Auftrag der TU Berlin weiterentwickelt.

5.1 Technologische Realisierung

Es gibt heute von einigen großen Software- und Systemanbietern [33, 64, 90] zahlreiche Lösungen im Bereich Identity Management. Auch im Open Source Bereich sind Lösungen zu finden [3, 93]. In verschiedenen Forschungsprojekten wurden unter Verwendung von freier Software und von Eigenentwicklungen Werkzeuge zur Umsetzung eines Autorisierungsmanagements [7, 30, 47] entwickelt.
Der folgende Abschnitt spiegelt einen möglichen Lösungsansatz für die technische Umsetzung wider. Er orientiert sich an der zur Zeit an der TU Berlin eingesetzten Infrastruktur. Da technologische Details des Systems immer wieder den Änderungen von Infrastruktur und Prozessen angepasst werden müssen, wird auf die Darstellung von Implementierungsdetails verzichtet.

5.1.1 Webschnittstelle / Benutzungsschnittstellen

Der Einsatz von WWW-basierten Benutzungsschnittstellen ist im IDM-Bereich, wie auch bei vielen anderen Anwendungen, ein Quasistandard. Eine Webschnittstelle bietet für Benutzer und Anbieter in diesem Bereich eine Reihe von Vorteilen: Plattformunabhängigkeit, Verteiltheit, minimale technische Voraussetzungen, Nutzung von Synergieeffekten, geringer

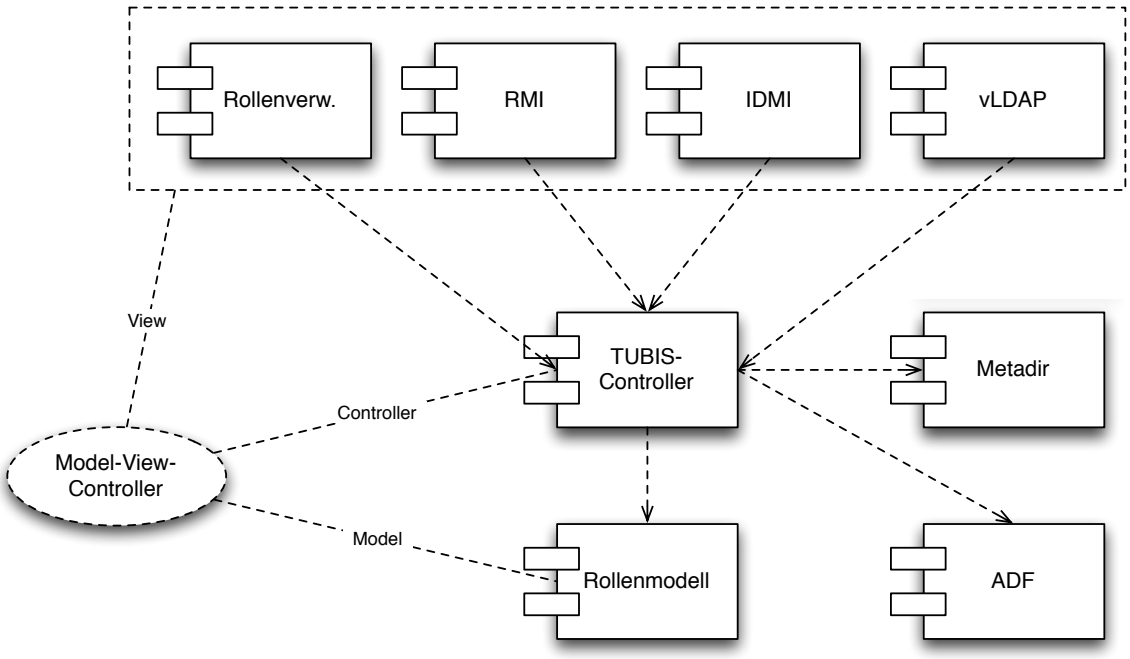

Abbildung 5.2: Komponentendiagramm: Model-View-Controller-Architektur des TUBIS-Kerns

Schulungsaufwand, hohe Akzeptanz, Barrierefreiheit, Portalintegration (alle Anwendungen an einem Ort). Der Nachteil der eingeschränkten Gestaltungsmöglichkeiten wird heute durch Techniken wie AJAX relativiert. Im Kontext der TU Berlin werden verschiedene WWW-basierte Anwendungen angeboten, die den Lebenszyklus einer Identität fast vollständig abdecken. Allein das Löschen einer Teilidentität inklusive der Löschung aller damit verknüpften Daten geschieht automatisiert ohne Benutzungsschnittstelle. Der Prozess wird durch Exmatrikulation, Ende der Vertragslaufzeit oder Ablaufdatum angestoßen und berücksichtigt diverse Fälle, in denen der Prozess ausgesetzt werden kann (z.B. bei Vertragsverlängerung oder Folgeverträgen).

5.1.2 Model-View-Controller Architektur

Das Rollenverwaltungssystem TUBIS basiert auf einer Model-View-Controller Architektur (Abb. 5.2). Die Model-View-Controller Architektur wird häufig bei interaktiven Anwendungen eingesetzt. Dabei wird das System in drei Komponenten gegliedert. Das Modell enthält die Kernfunktionalität und die Daten, auf denen gearbeitet wird. Diese Daten können von unterschiedlichen Ansichten (engl. Views) präsentiert werden. Die Kontrollkomponente (engl. Controller) ist für die Verarbeitung der Eingaben zuständig. In TUBIS werden die Daten aktuell über eine Persistenzschicht in einer relationalen Datenbank abgelegt. Zu Beginn der TUBIS-Entwicklung wurde das Datenmodell in einer Dateistruktur direkt im Dateisystem abgelegt und über eine Berkeley DB [72] indiziert. Ziel dieser Technik war u.a., mit Hilfe einer RBAC-Schicht im Betriebssystem bereits einen rollenbasierten Zugriff auf die persistenten Objekte auf dem Massenspeicher zu erreichen. Dieses Vorhaben wurde zugunsten einer

schnelleren Weiterentwicklung später aufgegeben.

Die eigentliche Programmlogik befindet sich im Controller. Dieser enthält Metadirectoryfunktionen, Zugriffslogik und Nebenbedingungskontrollen für das Rollenmodell sowie die eigentliche Zugriffsentscheidungsinstanz (ADF). Der Controller ist in Form einer JAVA-Bibliothek implementiert, die von den "View"-Komponenten genutzt wird. Zur Zeit erfolgt die Überführung in eine JEE-Architektur, die zukünftig eine flexiblere Kontrolle bei Nebenläufigkeiten sowie eine bessere Skalierbarkeit erwirken soll.

5.1.3 Alternative View-Komponenten

Die Webschnittstelle stellt die *primäre* "View"-Komponente dar. Andere Schnittstellen zur Darstellung des Modells sind ferner die Rollenabfrageschnittstelle (Role Management Interface, RMI) und die Identätsschnittstelle (IDMI) sowie die virtuelle LDAP-Schnittstelle.

Jython als Interpreter-Schnittstelle

Über eine Jython-Integration ist es möglich, API-Funktionen aus einer Kommandozeile aufzurufen oder Scripte zu entwickeln, um Routineaufgaben bei der RBAC-Modellverwaltung zu automatisieren. Diese Schnittstelle ist bestens geeignet, um bislang nicht vorgesehene aber selten gebrauchte oder gar einmalige Anfragen in das Modell zu stellen, Fehleranalysen im Modell durchzuführen, für administrative Zugriffe bei schlechter Netzanbindung oder bei wiederkehrenden Operationen, die über eine Skriptsprache leicht parametrisiert werden können.

Die Kombination Jython/Java oder auch Beanshell/Java stellt eine Möglichkeit dar, wie Nebenbedingungen im RBAC-Modell in Form von Programmausdrücken im Objekt gespeichert und zur Laufzeit von einem Interpreter ausgewertet werden können. Dabei ist darauf zu achten, dass den Objekten keine Methoden zur Verfügung gestellt werden, die eine Privilegieneskalation ermöglichen.

Der Controller als Java-API

Der Controller kann selbst als Java-API genutzt werden. Er wird als Bibliothek zur Verfügung gestellt, die von View-Komponenten genutzt werden kann. Dies erfordert, dass in der Controller-Logik jeweils auf Nebenläufigkeiten geachtet werden muss, bietet jedoch eine sehr effiziente Anbindung an den Controller.

5.1.4 Einsatz der Webanwendungen

Das Provisioning, die Selbstverwaltung, das personalisierte Portal, die Rollen-, Gäste- und Externenverwaltung sind jeweils als Webanwendungen realisiert. Sie werden im Folgenden kurz vorgestellt.

Provisioning

Das Provisioning findet für alle Benutzergruppen, mit Ausnahme der Gäste, in folgenden drei Schritten statt:

1. Der Benutzer erhält ein Anschreiben, in dem der Prozess beschrieben wird und der einen Barcode enthält.

2. Mit dem Anschreiben wird der Benutzer bei der so genannten Kartenausgabestelle mit Chipkarte, TAN-Liste und einem Initialpasswort ausgestattet. Nahe der Ausgabestelle befinden sich Fotoautomaten, mit denen man unter Verwendung des Barcodes auf dem Anschreiben Fotos erstellen kann. Der Barcode dient beim Drucken des Ausweises (Sichtfunktion der Chipkarte) der Zuordnung des Fotos sowie dem Auslesen der vor Ausstellung des Anschreibens von der zuständigen Stelle erfassten Stammdaten.

3. Mit dem Initialpasswort kann eine Webseite zur Erstellung eines Benutzerkontos aufgerufen werden (s. Abb. 5.3). In diesem Zusammenhang werden gemäß der Vorgaben der Organisation Benutzerkontoname, Passwort für den Zugang sowie E-Mailadresse gesetzt.

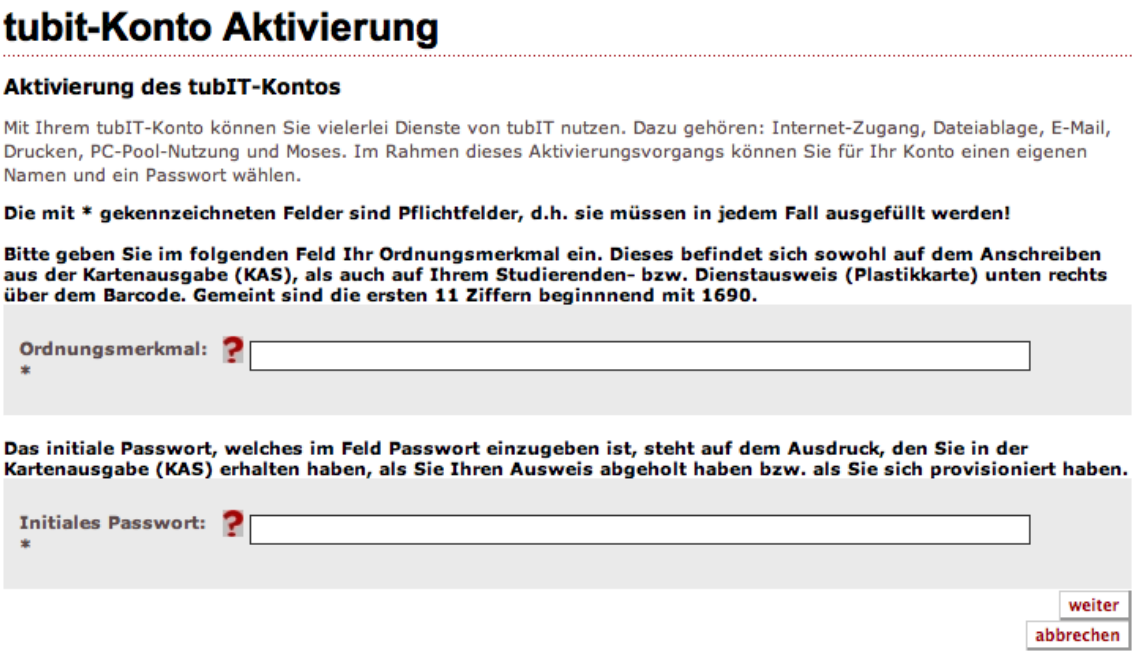

Abbildung 5.3: Bildschirmfoto der Kontoaktivierung

Diese Schritte haben in den Hintergrundsystemen die folgenden Änderungen zur Folge:

- Die Person wird im Rollensystem angelegt, Position und damit Zuordnung zur Organisationseinheit sowie initiale Rollen werden an Hand der Informationen aus den Primärdatenquellen zugewiesen.
- Bei der Freischaltung des Benutzerkontos werden evtl. existierende Altkonten abgewählt, um Missbrauch mit diesen Konten zu verhindern. Benutzername, Passwort und Attribute, wie die E-Mailadresse, werden in den Verzeichnisdiensten eingetragen.
- Massenspeicherbereiche im SAN sowie das E-Mailkonto werden angelegt.

Eine Herausforderung beim Provisioning stellt die Beachtung aller auftretenden Spezialfälle dar. Für diese Fälle sollte geprüft werden, ob eine Implementierung nötig ist oder ob es organisatorisch möglich ist, den Spezialfall auf bereits implementierte Fälle abzubilden. Viele dieser

Fälle sind historisch entstanden und können nach Prüfung der aktuellen Sachlage abgelöst werden.

Selbstverwaltung

Selbstverwaltung beschreibt die Benutzungsschnittstellen (oder oft, wie auch an der TU Berlin, die Sammlung von Benutzungsschnittstellen) für die Verwaltung der eigenen personenbezogenen Attribute, der technischen Attribute, die zu einer Person gehören sowie technisch/administrative Prozesse, die im Dialog selbst durchgeführt werden können.

Abbildung 5.4: Bildschirmfoto des Abschnitts "Persönliche Daten" aus dem personalisierten TU Portal

Abb. 5.4 zeigt die Auswahl der unterschiedlichen Programmmodule, die unter dem Begriff "Selbstverwaltung" zusammengefasst werden können. Aus technischer Sicht handelt es sich dabei im Einzelnen um folgende Realisierungen:

Personaldaten: Zugriff auf ein Webportal eines Fremdherstellers, das in das Portal integriert wurde.

Konto und Rollen: Webschnittstellenmodul des selbst entwickelten rollenbasierten IAM-Systems TUBIS.

Kennwort ändern: Ein Script zur synchronen Passwortänderung unter Berücksichtigung aller verwendeten Verzeichnisdienste.

TAN Liste: Eine Anwendung zur Verwaltung der indizierten Einmalpasswörter.

Das personalisierte Portal

Das Portal der TU Berlin wird durch ein CMS dargestellt. Für die Darstellung einer personalisierten Startseite mit Anwendungen, für die die authentisierte Person Rollen besitzt, wurden einige Scripte erstellt, die in die CMS-Seite eingebettet wurden. Abhängig von den Berechtigungen der Person werden die Menüpunkte in der Seitenleiste links (Abb. 5.5) aufgebaut.

Ferner verbirgt sich hinter dem Webportal eine Webföderation. Eine Authentisierung gegenüber dem Portal ist nur einmal pro Sitzung notwendig und kann mit verschiedenen Authentisierungsmethoden erfolgen. Mehr zur Webföderation folgt im Kapitel 5.2.1.

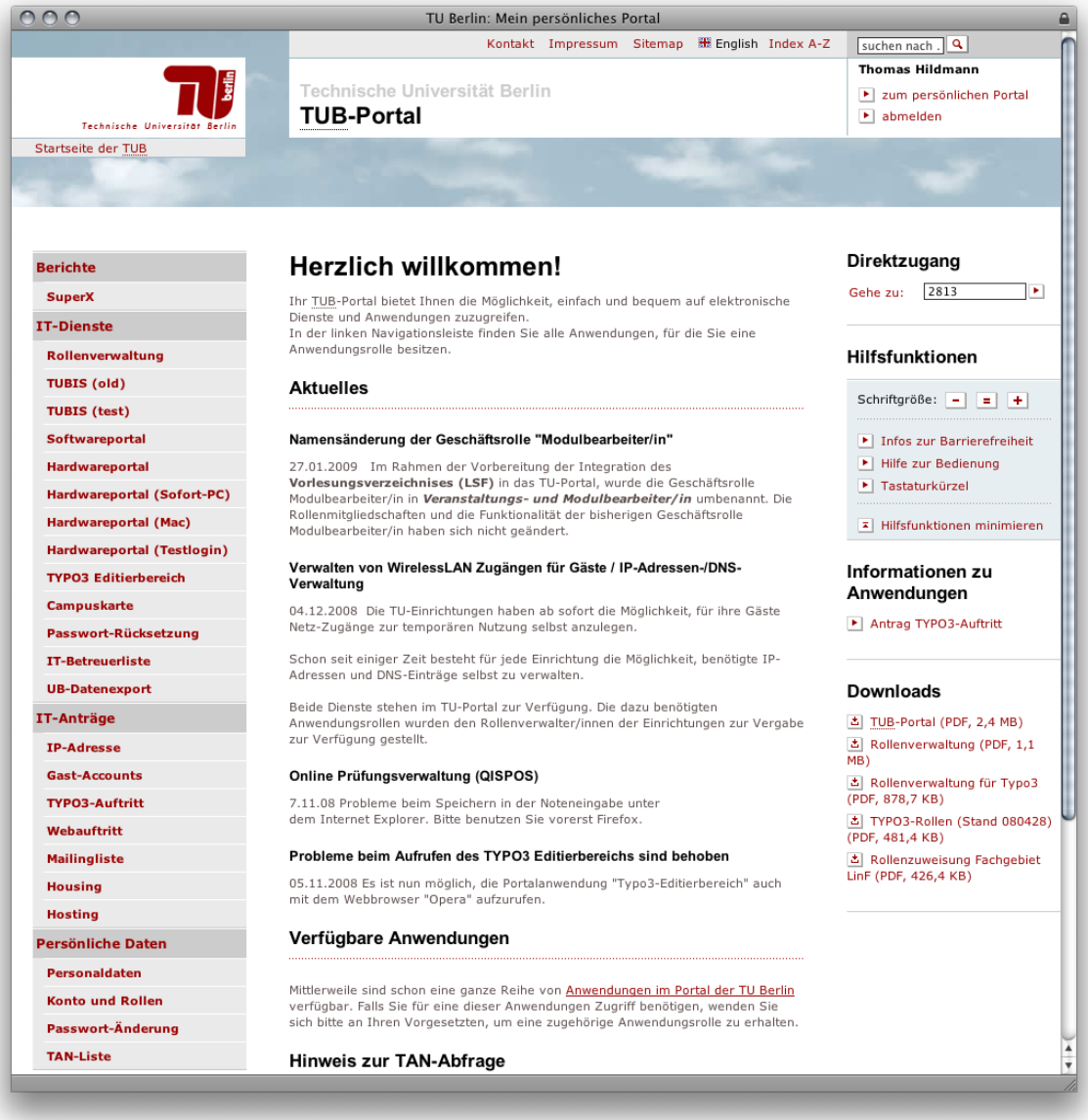

Abbildung 5.5: Bildschirmfoto des personalisierten Portals

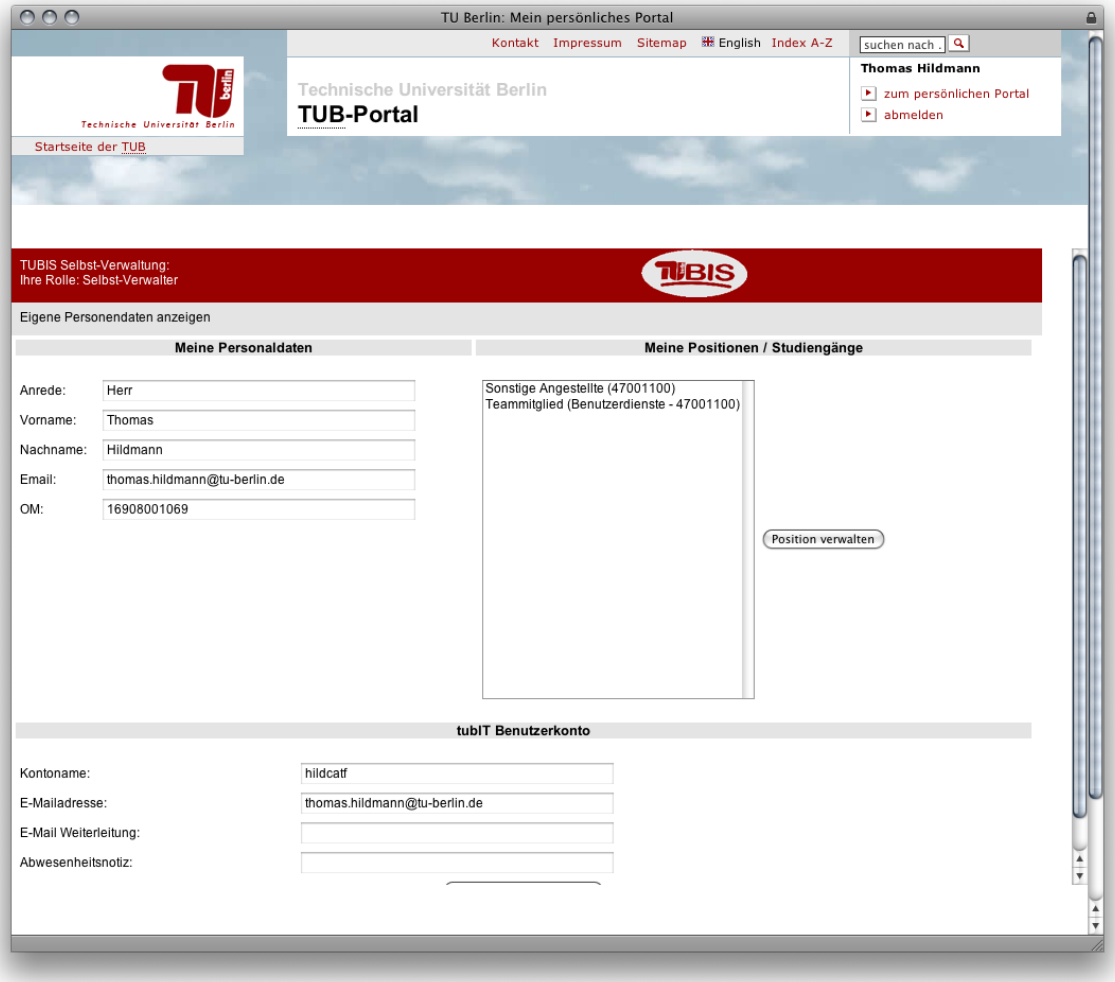

Abbildung 5.6: Bildschirmfoto des Startbildschirms des Selbstverwaltungsmoduls "Konto und Rollen"

Die Rollenverwaltung

Die Rollenverwaltung ist modularisiert und besitzt unterschiedliche Sichten. Mitarbeiter und Studierende verfügen nach Anmeldung am TU Portal über Zugriff auf verschiedene Selbstverwaltungsfunktionen. Darunter befindet sich auch der Punkt "Konto und Rollen" (Abb. 5.6). Über die Auswahl einer Position oder einer Teammitgliedschaft gelangt der Benutzer schließlich zur Ansicht und Verwaltung der eigenen Geschäftsrollen (Abb. 5.7).

Verwaltern einer Organisationseinheit steht zusätzlich eine erweiterte Benutzungsschnittstelle zur Verfügung, die das Definieren von Geschäftsrollen, sowie das Zuweisen von Rollenmitgliedschaften für Rollen der eigenen Organisationseinheit ermöglicht. Dienstanbietern steht eine Anwendungsverwaltung zur Verfügung, mit der es möglich ist, Rollen für Personengruppen oder Organisationseinheiten zur Verfügung zu stellen.

Abbildung 5.7: Bildschirmfoto von "meine Geschäftsrollen"

Gäste und Externenverwaltung

Gäste sind Personen, die nur für sehr kurze Zeit an der TU Berlin sind, wie das bei Tagungen, Projektbesuchen, beauftragten Fremdfirmen, Gastvorträgen etc. der Fall ist. Diesen Personen wird eine Netzwerkverbindung in Form eines WLAN-Kontos zur Verfügung gestellt.

Gastkonten werden von Einrichtungsleitern oder von ihnen beauftragten Personen angelegt. Für diese steht eine Webapplikation zur Verfügung (s. Abb. 5.8), mit der für einen gegebenen Zeitraum die vorgegebene Zahl von Gastkonten erstellt werden kann. Die Anwendung erstellt pseudonyme Benutzernamen mit zufälligen Passwörtern. Die Liste wird dem Benutzer zur Verteilung übermittelt.

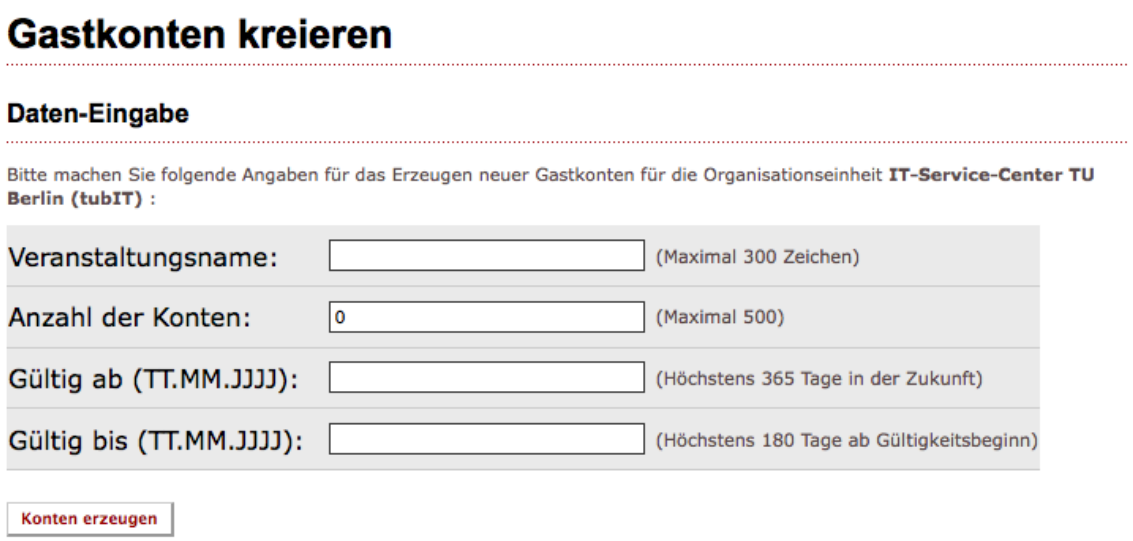

Abbildung 5.8: Bildschirmfoto der "Gastverwaltung"

Gäste werden nicht in das Rollenmodell aufgenommen. Die Konten werden auf dem Hintergrundsystem für das WLAN angelegt. Nach der Veranstaltung werden die Konten automatisch gelöscht.

Unter "Externe" werden Personen geführt, die weder von der Personal- noch von der Studierendenverwaltung an der TU Berlin administriert werden. Hierzu gehören z.B. Nebenhörer, Mitarbeiter mit Werkverträgen oder Gastdozenten.

Zielgruppe der Externenverwaltungsanwendung sind auch hier Leiter von Einrichtungen oder von ihnen beauftragte Personen. Die Daten der Externen werden in einer vom Rechenzentrum verwalteten Datenbank gehalten.

Über die Webschnittstelle (s. Abb. 5.9) können die Stammdaten sowie die Vertragslaufzeit erfasst werden. Am Ende des Vorgangs wird ein PDF generiert, dass ausgedruckt werden kann und das als Anschreiben im Sinne des Provisioning dient (siehe Abschnitt "Provisioning"!).

An der TU Berlin hat sich folgende Regel für ein umfassendes Autorisierungsmanagement bewährt: Bei der Etablierung des Systems werden so viele bereits vorhandene Datenquellen wie möglich genutzt. Fehlende Quellen werden durch den Aufbau neuer Datenbestände hinzufügen.

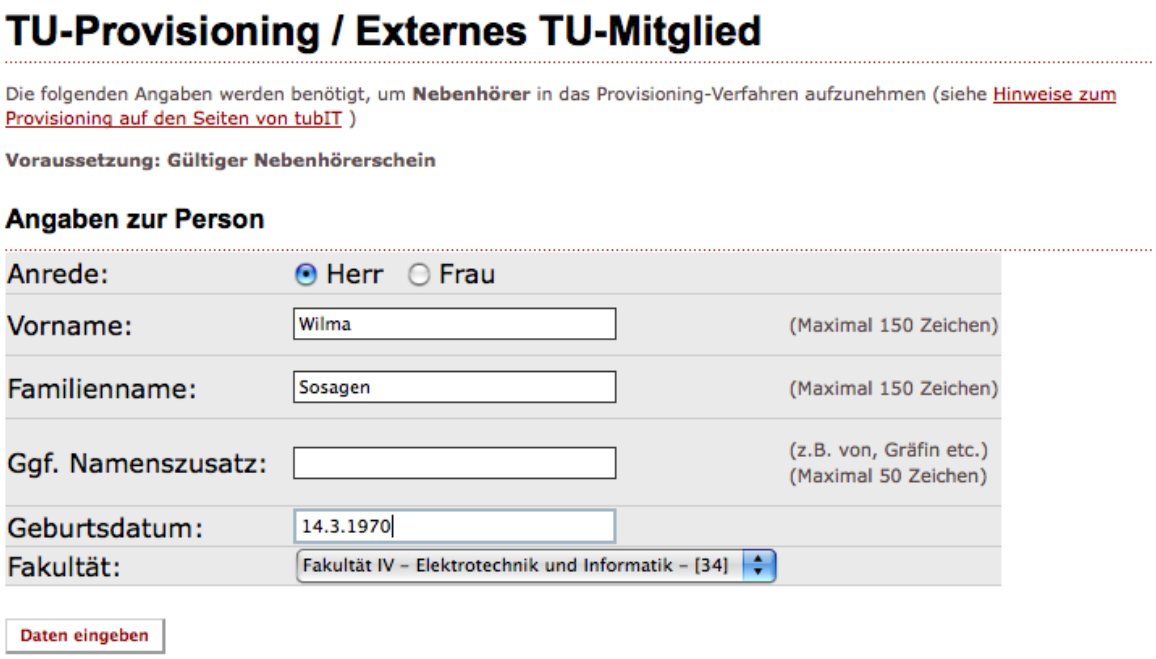

Abbildung 5.9: Bildschirmfoto der "Externenverwaltung"

Modularisierung in Webelemente

Zur Zeit arbeiten wir an der TU an einer Modularisierung der Webschnittstellen, die das Einbinden einzelner Elemente der Rollenverwaltung in andere Webseiten ermöglicht. Damit soll die Trennung "Anwendung X" auf der einen und "Rollenverwaltungsprogramm" auf der anderen Seite aufgehoben werden. Einige Prozesse, wie z.B. die Erstellung eines Webauftritts, erfordern eine Kombination aus Konfiguration in der Anwendung und entsprechende Rechtekonfigurationen im Rollenmodell. Die Integration der Anwendungen in die AA-Infrastruktur wird mit der Einbettung der Rollenverwaltungselemente perfektioniert. Im Ergebnis versprechen wir uns von diesem Schritt eine Reduzierung von Blockaden in Arbeitsabläufen, eine verbesserte Transparenz der Abhängigkeiten zwischen Zugriffskontrolle und Prozessen und ein noch besseres Verständnis der Benutzer in Bezug auf das Autorisierungsmanagement und damit in letzter Konsequenz eine Erhöhung der Sicherheit.

Die Einbindung von RBAC-Elementen in Anwendungen setzt ein gutes Verständnis und die Zusammenarbeit von Anwendungsentwicklern der verschiedenen Projekte voraus. Neben der Schulung der Softwareentwickler in Bezug auf die TUBIS-GUI-API und den zugrunde liegenden Konzepten besteht eine besondere Herausforderung auch darin, das Verhalten der Anwendungen zu synchronisieren und aus Benutzersicht den Anschein eines einzelnen Systems zu erwecken. Die Arbeiten an diesem Projekt werden daher noch einige Zeit in Anspruch nehmen.

5.1.5 Authentisierung

Jeder Autorisierung in IT-Systemen muss eine Authentisierung vorangehen. Zugriffsentscheidungen werden jeweils gemäß der ermittelten Teilidentität des Subjekts getroffen. Die Au-

thentisierung kann von der Autorisierung entkoppelt werden. Die Identität kann via Berechtigungsnachweis oder über Rückfragemechanismen ermittelt werden. In vielen heute am Markt befindlichen Produkten wird die Autorisierung der Authentisierung gleich gesetzt. Die Existenz eines Benutzerkontos ist in diesem Fall gleichbedeutend mit der Berechtigung der Nutzung.

Die Authentisierung sollte für Benutzer und Anwendung transparent sein, d.h. unabhängig von Protokoll und Methode. Ein Benutzer sollte eine Anwendung in gleicher Weise benutzen können, gleich ob er sich mit einer Chipkarte oder einem biometrischen Verfahren beim System authentisiert hat. Unterscheidungen können ggf. in Bezug auf die Sicherheit der Verfahren gemacht werden. So kann beispielsweise unterschieden werden, ob sich ein Benutzer mittels Benutzername und Passwort oder über ein kryptografisches Verfahren angemeldet hat. Die Authentisierung kann verschiedene Methoden zur Verfügung stellen, um das jeweilige Sicherheitsniveau zu erreichen und auf der anderen Seite jeweils das größte Maß an Benutzungsfreundlichkeit zu bieten. Ein universell einsetzbares Authentisierungsmodul ist eine Möglichkeit, um den Benutzern die Verwaltung ihrer Zugänge zu vereinfachen (z.B. ein Passwort für alle Anwendungen).

Einen Schritt weiter geht die Forderung nach einem Single-Sign-On Mechanismus. Dieser wird an der TU Berlin für alle Webanwendungen mit Hilfe eines Ticket-Mechanismus und der Verwendung von Proxy-Servern umgesetzt.

Authentisierungsprotokolle

Hinsichtlich der Authentisierung müssen die Anwendungen bezüglich ihrer Authentisierungsschnittstelle unterschieden werden. So gibt es Dienste, die folgende Authentisierungstechniken unterstützen:

- Radius (Remote Authentication Dial-In User Service)
- Kerberos
- LDAP-Bind / LDAP-Query
- webbasierte Anwendungen

Das Radius-Protokoll kommt sowohl bei der Einwahl von Benutzern über Modem/ISDN-Leitungen, wie auch bei der Authentisierung von WLAN-Clients zum Einsatz. Eine PKI-basierte WLAN-Authentisierung wäre denkbar, ist jedoch technisch sehr viel anspruchsvoller.

Das Kerberos-Protokoll wird für den Rechnerzugang von Windows- bzw. Linux-Systemen genutzt sowie zur Authentisierung von Netzwerkdateisystemnutzern. Ferner kann eine LDAP-`bind()` Anfrage über Kerberos realisiert werden, was nicht nur einen Kerberos-basierten Schutz des LDAP-Servers zur Folge hat, sondern ebenfalls die Nutzung von Kerberos für jeden Dienst, der auf LDAP-Bind basiert (z.B. SMTP-, IMAP-, POP3-Server). Die Kerberos-Konten werden an der TU Berlin im Rahmen des Provisioning angelegt und bei Passwort-Änderungen jeweils synchronisiert. Um sich gegen Kerberos-basierte Dienste auch mit Hilfe von Smartcards authentisieren zu können, ist die Nutzung des bereits vorhandenen Active Directories (AD) als Kerberos-Server geplant, da dieser bereits in der Lage ist, PKI-basierte Authentisierungen durchzuführen.

Neben den Anwendungen, die eine Benutzeranmeldung über LDAP-Bind prüfen, gibt es auch solche, die bestimmte Attributanfragen an einen LDAP-Server stellen. Solche Anwendungen können am besten über ein virtuelles Verzeichnis angebunden werden, um z.B. zu verhindern, Passwörter im LDAP speichern zu müssen.

Campuskarte Version 2 (CKv2)

Das Authentisierungsrahmenwerk eines umfassenden Systems sollte in der Lage sein, verschiedene Authentisierungsmethoden zuzulassen und für die integrierten Anwendungen eine einheitliche Schnittstelle zur Verfügung zu stellen. Gängige Methoden zur Identifizierung von Benutzern sind in 2.2.2 beschrieben.

An der TU Berlin werden zur Zeit zwei Smartcard-Varianten unterstützt:

Die Campuskarte v1 ist eine Javacard-basierte Smartcard, die ein Authentisierungsschema unterstützt, das die digitalen Spuren, die bei der Nutzung einer solchen Karte zwangsläufig entstehen, gemäß der Methoden der mehrseitigen Sicherheit verteilt und damit eine pseudonyme Nutzung fördert [44] (s. Kapitel 3.5.1). Die Pflege einer solchen Eigenentwicklung bindet jedoch Ressourcen, wogegen die Vorteile einer pseudonymen Authentisierung den Nutzern zur Zeit nur schwer zu vermitteln sind, so dass diese Lösung zugunsten einer standardkonformen Lösung, die unmittelbar für ein breites Spektrum an Plattformen zur Verfügung steht, aufgegeben wurde.

Bei der aktuellen Version der Campuskarte handelt es sich um eine handelsübliche Signaturkarte, mit der eine SSL-Client-Authentisierung durchgeführt werden kann. Dies vereinfacht die Infrastruktur erheblich. Ferner müssen keine Mittel mehr für die Pflege der Middleware, für das aufwändige Authentisierungsverfahren bereitgestellt werden. Dafür steht die Middleware nun für verschiedene Plattformen (Windows XP, Vista, MacOS X Leopard, Linux, Solaris) zur Verfügung. Aufgegeben werden musste hierdurch die Trennung von Card-Identifier (CID) und Ordnungsmerkmal (OM). Auf eine CID wurde bei der CKv2 verzichtet. Zur Sperrung einer Karte wird das entsprechende Authentisierungszertifikat zurückgezogen. Dieses enthält das OM im Zertifikatfeld "Subject". Da der Webproxy jeweils über das Dekorator-Muster die Anwendungen mit Informationen versorgt, kann noch immer gesteuert werden, welche Anwendung welche personenbezogenen Daten erhält. Die Verteilung der Informationen wurde bei der CKv1 vornehmlich dadurch erreicht, dass jede Anwendung einen unabhängigen Webproxy bekam. Diese Trennung wurde zugunsten einer Lastverteilung und Redundanz aufgegeben. Bei gleicher Architektur könnte eine Verbesserung durch eine Protokollierung, wie in Kapitel 3.5.4 beschrieben, erreicht werden. Allerdings fallen immer vollständige Bewegungsdaten an den zentralen Webproxy-Servern an. Verzichtet man auf eine Verteilung des Dienstes, bleiben geeignete technische und organisatorische Maßnahmen, um das Anlegen von Protokolldateien zu verhindern.

5.1.6 Firewall/Proxy-Architektur und Verwendung des Dekorierer-Musters

Die Proxy-Architektur, in der eine sog. AEF den Zugriff zu einem Ziel frei gibt, wenn eine ADF den Zugriff gestattet, findet man in [1].

Die Nutzung einer Proxy-Architektur (Abb. 5.10) im Sinne einer AEF ist ein bewährtes Mittel, das häufig in Firewallprodukten eingesetzt wird. Für die Implementierung einer webbasierten AEF kann auf gebräuchliche Open Source Produkte zurückgegriffen werden. Das eigentliche AEF Modul besteht aus einem Servlet, das als eine Pipes-and-Filter Architektur

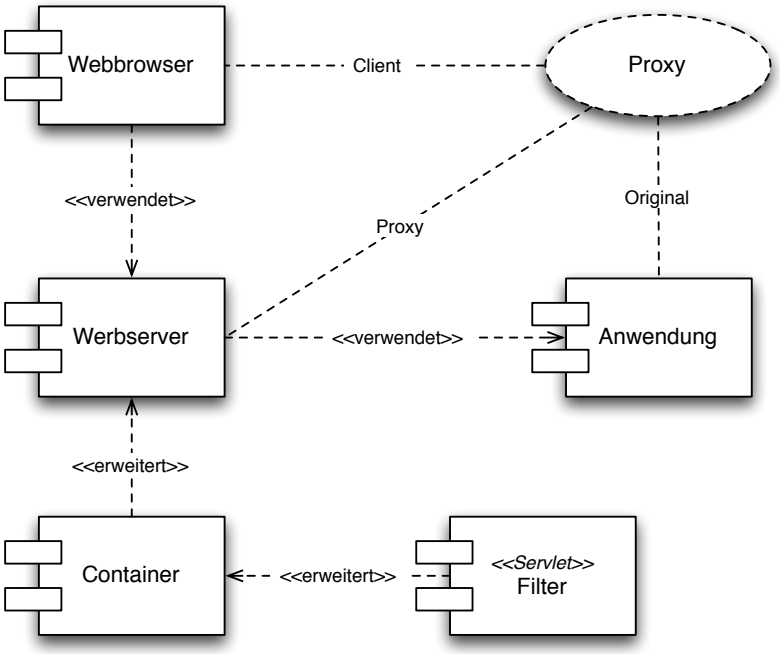

Abbildung 5.10: Komponentendiagramm: Proxy-Architektur der AEF

beschrieben werden kann. Es arbeitet nach dem Prinzip, nur solche HTTP-Anfragen weiterzuleiten, die von einer autorisierten Quelle kommen. Nicht authentisierte Clients werden auf eine Authentisierungs-URL umgeleitet und somit zur Authentisierung gezwungen. Ist der Client authentisiert, so wird die Autorisierung geprüft; ggf. wird dem Benutzer hierbei eine Rollenauswahl von sich ausschließenden Rollen präsentiert. Der Nutzer muss dann die Rolle wählen, in der er auf die Anwendung zugreifen will. Ist der Benutzer authentisiert und autorisiert, so wird die HTTP-Anfrage zum eigentlichen Webserver durchgelassen. Im einfachsten Fall ist die Zwischenschaltung des AEF-Proxy für die geschützte Webanwendung transparent.

Um die geschützte Webanwendung mit Rollen-, Identitäts- und SoD-Informationen (z.B. "Besteller *für Kostenstelle 47001100*") versorgen zu können, wird auf das Dekorator-Muster zurückgegriffen. Das Servlet ist in der Lage, die HTTP-Anfragen mit Parametern anzureichern. So können der Anwendung die Benutzerkennung, der Rollenname sowie weitere benötigte Informationen zur aktuellen Sitzung als Parameter übergeben werden. Dabei wird in der AEF das Prinzip verfolgt, im ersten Schritt alle HTTP-Parameter zu löschen, die in der Dekorator-Schnittstelle verwendet werden, um Fälschungen zu verhindern. Im nächsten Schritt werden die Variablen dann mit den Daten angereichert, die aus dem Rollensystem abgefragt werden.

Die Maschine, auf dem die Proxy-Architektur installiert ist, wird Authentisierungs-/Autorisierungs-Gateway (AAGW) genannt. Sie verfügt über drei Netzwerkschnittstellen (Abb. 5.11). Dabei ist eine über eine Lastverteilung mit dem Internet verbunden, die zweite hat eine Verbindung zum Netzwerk mit den zu schützenden Anwendungen und die dritte Schnittstelle führt in das Netz mit der Authentisierungs-/Autorisierungsinfrastruktur, in dem sich LDAP-Server, ADF und weitere in diesem Fall nicht benötigte Systeme, wie Kerberos, Active Directory etc. befinden.

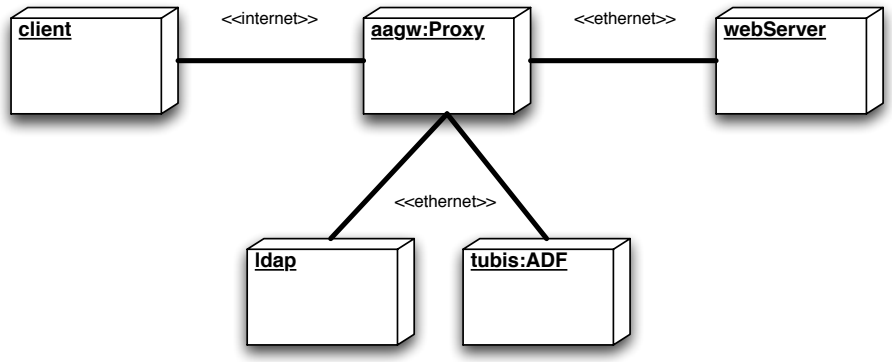

Abbildung 5.11: Verteilungsdiagramm: AAGW

Die AEF-Komponente ist ein wesentlicher Bestandteil der AAI und dient der Realisierung unterschiedlicher Authentisierungsmethoden im Web sowie der Implementierung des SSO-Mechanismus.

5.1.7 Verwendung von Webservices

Webservices (`http://www.w3.org/2002/ws/`) kommen als Basistechnologie für die Erstellung verteilter Anwendungen zum Einsatz. Die Anbindung von unterschiedlichen Anwendungen auf unterschiedlichen Plattformen und mit variierenden Programmiersprachen ist über Webservices möglich.

TUBIS stellt drei Arten von Schnittstellen nach außen zur Verfügung:

1. Einen Push-Service, der z.B. Verzeichnisse wie LDAP, Kerberos und das Active Directory sowie Datenbanken versorgt.

2. Den virtuellen Verzeichnisdienst, der auch Echtzeitrolleninformationen enthält.

3. Eine Reihe von Webservices zur internen und externen Kommunikation.

Insbesondere werden die folgenden Schnittstellen innerhalb der AAI über Webservices realisiert:

Provisioning: Über einen Webservice wird das IDM angewiesen, die Stammdaten für ein neues Mitglied anzulegen und die korrespondierenden Daten mit der Primärquelle zu verknüpfen. Ferner wird in diesem Zusammenhang ein initialer Satz von Rollen zugewiesen. Ergebnis des Methodenaufrufs sind die für die Erstellung von Anschreiben und Ausweisen benötigten Informationen.

Rollenabfrage: Ergebnis der Authentisierung im AAGW ist eine eindeutige Benutzerkennung. Mit dieser kann über ein Webservice abgefragt werden, für welche Anwendungen Rollen zur Verfügung stehen und bezogen auf eine Anwendung, um welche Rollen es sich handelt. Diese Informationen werden zum Aufbau des personalisierten Portals und bei der Autorisierung bezüglich einer Anwendung benötigt.

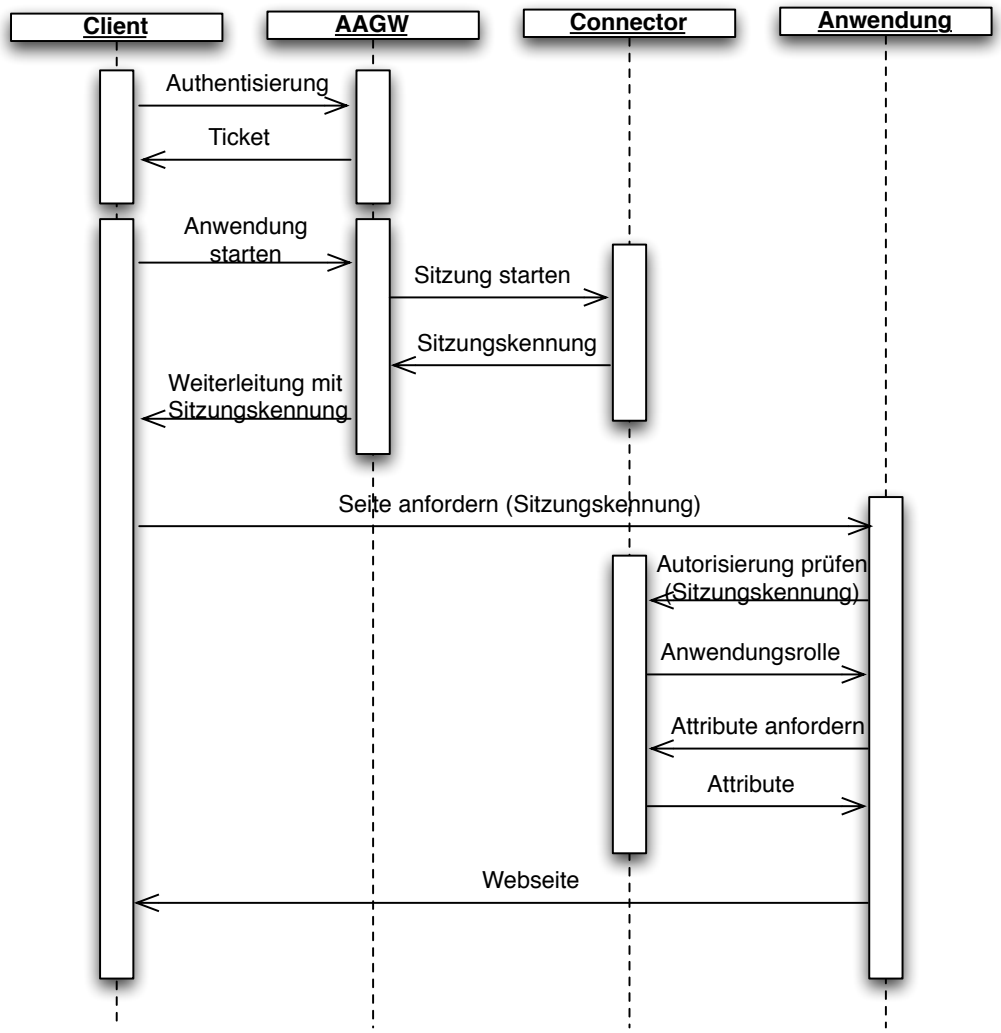

Abbildung 5.12: Sequenzdiagramm: Integration über Connector

Identitätsinformationen: Einige Anwendungen benötigen zusätzliche Attribute zu einer Person. Eine Möglichkeit zur Abfrage dieser Attribute besteht über einen Webservice, der u.a. vom Dekorator im AAGW benutzt wird, um die Autorisierungsdaten mit benötigten Informationen anzureichern.

Brücken: Für verschiedene Anwendungen werden spezialisierte Webservices verwendet, die Methoden zur Einbettung in die AAI zur Verfügung stellen. Dabei wird ein Teil vom AAGW benutzt, um eine Sitzung zu öffnen (Abb. 5.12). Die Sitzungskennung wird dann an die Anwendung übergeben, die sich dann mit dieser Kennung an den korrespondierenden Connector wenden kann, um answendungsspezifische Informationen abzufragen oder Methoden aufzurufen.

5.2 Architektonische Realisierung

Das Bindeglied zwischen den organisatorischen Maßnahmen und der technischen Realisierung eines umfassenden Autorisierungsmanagements ist die Schaffung von architektonischen Voraussetzungen. Hierzu gehört der Aufbau eines Authentisierungs-/Autorisierungs-Rahmenwerkes und die Integration der Anwendungen in dieses Rahmenwerk. Eine entscheidende Voraussetzung für die erfolgreiche Umsetzung eines organisationsumfassenden Autorisierungsmanagements ist die Wahl eines geeigneten Modellierungsansatzes. Die Modellierung muss schließlich mit speziellen Werkzeugen abgestimmt werden. Die Konfiguration des Autorisierungsmodells muss in die Arbeitsabläufe eingebunden werden.

5.2.1 Authentisierungs-/Autorisierungs-Rahmenwerk

Aufgabe des AA-Rahmenwerkes ist die Entkopplung von Benutzeridentifikation und Autorisierung von den zu nutzenden Anwendungen. Ziel ist die Nutzung von zentralen Benutzer- und Autorisierungsinformationen für alle Anwendungen. Die Aufgabe lässt sich in Bezug auf Webanwendungen mit dem Aufbau einer Webföderation umschreiben. Diese Webföderation kann organisationsweit oder sogar organisationsübergreifend angelegt sein.

Das von meiner Forschungsgruppe entworfene AA-Rahmenwerk arbeitet nach den selben Prinzipien wie andere Rahmenwerke in diesem Bereich. Nach erfolgreicher Authentisierung mit einer der angebotenen Methoden gilt das Client-System als identifiziert und kann somit als Subjekt im Sinne der Zugriffskontrolle behandelt werden. Dieser Zustand wird im Webbrowser des Client-Systems durch das Setzen eines signierten Cookies repräsentiert. Während ein Anwendungs-Cookie alle für die Anwendung relevanten Attribute der Teilidentität des Subjekts enthält, ist in einem Domänen-Cookie lediglich die Benutzerkennung vermerkt sowie die die Sitzung betreffenden Daten, wie Start- und Verfallsdatum der Sitzung, zugehörige IP-Nummer etc. Wechselt der Benutzer die Anwendung innerhalb der Webföderation, kann das AAGW anhand des Domänen-Cookies erkennen, dass es sich um einen bereits identifizierten Benutzer handelt. Die Anwendung ist nun in der Lage, anhand der Benutzerkennung über die entsprechenden Webservice-Schnittstellen nicht nur die Autorisierungsinformationen, sondern auch die nötigen Attribute des Benutzers abzufragen. Ergebnis ist ein Anwendungs-Cookie für die aktuelle Anwendung.

Nutzung der pseudonymen Autorisierung

Das in Kapitel 3.5.3 beschriebene Verfahren zur Vermeidung von Bewegungsprofilen setzt eine Trennung der ADF in ADF_n und ADF^* voraus. Ferner muss ein Mixen-Netz betrieben werden, das vor allem dann wirkungsvoll ist, wenn es von verschiedenen Parteien betrieben wird. Dieser Aufwand steht in keinem Verhältnis zur Verschleierung der genutzten Anwendungen innerhalb einer Universität (vor allem vor dem Hintergrund, dass diese Daten im Webproxy bei der Authentisierung ebenfalls anfallen). Das Verfahren könnte jedoch vor dem Hintergrund einer organisationsübergreifenden AAI an Bedeutung gewinnen.

In [45] wird die Idee von gemeinsam genutzten Rollensystemen für kooperierende Organisationen beschrieben. Ein solcher Ansatz wäre für die Autorisierung von Nebenhörern oder für hochschulübergreifende Dienstleister interessant. Je nach Hochschulgröße könnte dabei eine Ausgliederung der Rollenverwaltung oder das Betreiben eines Knotens in einem Netzwerk verteilter Rollensysteme in Frage kommen.

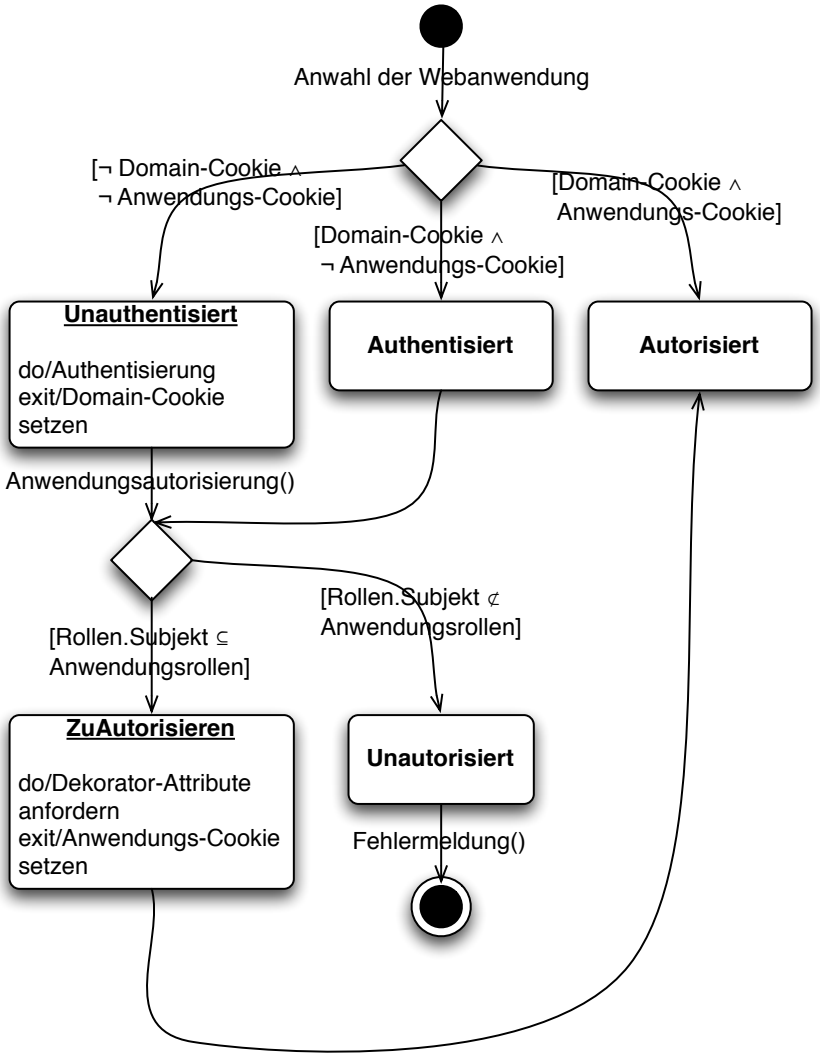

Abbildung 5.13: Zustandsdiagramm: Authentisierung/Autorisierung

5.2.2 Varianten der technischen Integration von Anwendungen

Die einfachste Art der Integration von Diensten in die Infrastruktur kann durch die Bereitstellung von standardisierten Schnittstellen wie LDAP, Active Directory, Radius, Kerberos, etc. realisiert werden. Allerdings dienen die bereits diskutierten Protokolle lediglich der Prüfung der Identität des Benutzers durch Authentisierung. Ein Zugriff kann jeweils nur gestattet oder abgelehnt werden. Die Autorisierung kann folglich nur sehr ungenau gesteuert werden. Evtl. ist eine weiterführende Autorisierung umsetzbar, wie z.B. im Fall einer Massenspeichernutzung, bei der der Zugriff auf einzelne Verzeichnisse oder Dateien definiert und somit eine Autorisierung gesteuert werden kann. Hier kann beispielsweise eine Rollenmitgliedschaft über Gruppenzugehörigkeiten oder ACLs abgebildet werden.

Das Hauptaugenmerk der folgenden Beschreibung liegt in der Integration von webbasierten Anwendungen. Die beschriebenen Techniken sind jedoch auf unterschiedliche Anwendungen und Protokolle übertragbar.

Nutzeridentifikation: Für einige Anwendungen wird einzig gefordert, dass eine vorherige Authentisierung des Benutzers durchgeführt wird. Beispiele hierfür sind "Meldungen des Tages des IT-Dienstleisters", "Zugriff auf interne organisationsweite Dokumente", "Einsicht in die eigenen Benutzerkontendaten" usw. Für webbasierte Dienste kann in diesem Fall wahlweise eine Basic-Authentication gegen einen LDAP durchgeführt werden oder die sehr viel stärkere PKI-basierte Authentisierung über SSL-Clientauthentisierung (idealer Weise mit Hilfe einer Smartcard).

Einfache Rollenautorisierung: Einige Anwendungen erfordern die Mitgliedschaft in einer bestimmten Rolle (z.B. "nur für Mitarbeiter", "nur für Leiter einer Organisationseinheit") oder die Auswahl einer Rolle, in der auf die Anwendung zugegriffen werden soll.

Rollenabbildung: Die Nutzung von Rollen oder Benutzergruppen wird von einigen Webanwendungen bereits vorgesehen. So sind in Content Management Systemen, wie Typo3, Plone oder Wordpress bereits Rollen wie "Redakteur" oder "Chefredakteur" definiert. In diesem Fall bietet es sich an, die Rollen des umfassenden Autorisierungsmanagments auf die Rollen in der Anwendung abzubilden und wie in diesem Fall durch Kontext/SoD-Informationen zu erweitern.

Vollständige AAI-Integration: Anwendungen, die für die Infrastruktur entwickelt werden oder für die Programmquellen und entsprechende Mittel zur Anpassung zur Verfügung stehen, können vollständig in die Authentisierungs/Autorisierungs-Infrastruktur eingebunden werden. Dabei kann das zentrale RBAC-System entweder dazu genutzt werden, die anwendungsspezifischen Rollen- und Benutzerinformationen zu liefern oder es kann das gesamte Rechtesystem auf die ADF der Autorisierungsverwaltung übertragen werden.

Bevor eine Anwendung im Portal genutzt werden kann, muss sie zunächst im Rollensystem modelliert werden. Im Rollenmodell wird hierzu ein Objekt angelegt, dass die Anwendung repräsentiert. Es ist dann minimal *eine* Anwendungsrolle zu definieren, um die Verknüpfung der Anwendung mit den Geschäftsrollen zu ermöglichen. Danach müssen die Anwendungsrollen zur Verfügung gestellt werden. Das geschieht entweder für Einheitentypen, bestimmte Einheiten oder für Standardrollen. Über diese können z.B. auch alle Mitglieder der Universität erreicht werden oder durch Bereitstellen der Rollen für bestimmte Organisationseinheiten.

Besitzt ein Benutzer mindestens eine Geschäftsrolle, an die mindestens eine Anwendungsrolle einer Anwendung gebunden ist, so taucht diese Anwendung im personalisierten Portal des Benutzers auf.

5.2.3 Daten Im- und Export

Alternativ zur Integration von Anwendungen über das in Abschnitt 5.1.6 beschriebene Proxy-Konzept können die Zugriffsinformationen zwischen Rollensystem und Zielsystem synchronisiert werden. So kann das Teilmodell z.B. auf die ACL der Anwendung abgebildet werden.

Beobachter/Adapter-Architektur zur Synchronsiation von Zugriffsrechten

Im Rollenmodell werden Beobachter definiert, die Änderungen an den Objekten registrieren. Für eine Zielanwendung können Objekte definiert werden, über deren Änderungen das Rollensystem informieren soll. Die Beobachter sind im Persistenzmodell des Rollensystems implementiert. Ändern sich Daten im Rollenmodell, wird damit die Synchronisation auf das Zielsystem ausgelöst.

Zur Synchronisation müssen die Zustandsänderungen des Rollensystems jeweils in das Modell des Zielsystems überführt werden. Hierzu dienen Adapter, die z.B. Rollenmitgliedschaften auf ACL-Einträge abbilden, Verzeichnisse anlegen usw. Diese Adapter werden häufig Agenten genannt. Zielschnittstellen, die von den Agenten bedient werden, können z.B. LDAP, SQL-Anfragen, CSV- oder XML-Dateien sein.

Bei der Integration von Anwendungen mit Hilfe von Zustandssynchronisationen treten häufig die folgenden Probleme auf:

- Die Anwendungen sind so ausgelegt, dass sie selbst Änderungen am eigenen Zugriffsmodell vornehmen können. Für eine vollständige Integration müsste eine Zwei-Wege-Synchronsiation realisiert werden. Die Zugriffsmodelle lassen sich jedoch meist nicht bijektiv abbilden. In solchen Fällen werden Änderungen innerhalb der Anwendung dann durch das zentrale Rollensystem wieder überschrieben oder die anwendungsspezifische Modellierung existiert neben der des umfassenden Autorisierungssystems, was Inkonsistenzen und Seiteneffekte zur Folge haben kann.

- Vielen Zielanwendungen fehlt ein Benachrichtigungsmechanismus, der die Anwendung über Änderungen im Zugriffsmodell informiert. Dies kann im schlimmsten Fall zu inkonsistenten Zuständen in der Anwendung führen. Oft lässt sich dieses Problem nur durch eine Abschaltung der Anwendung während der Synchronisation verhindern.

Anbindung an Primärdatenquellen

Um Synchronisationsprobleme in Bezug auf die Primärdatenquellen zu vermeiden und im Sinne der Umsetzung des Prinzips der Datenvermeidung, können Daten jeweils bei Bedarf aus den Datenbanken bezogen werden, wenn sie gerade benötigt werden.

Einige Änderungen in Primärdaten lösen Modelländerungen im RBAC-System aus. Hierzu gehören z.B. Änderungen an der Organisationsstruktur oder das Umsetzen einer Person in eine andere Struktureinheit. Um einen regelmäßigen Abgleich der Daten mittels vollständiger Iteration zu vermeiden, werden bei der Implementierung an der TU Berlin lediglich die

Strukturinformationen alle 24 Stunden synchronisiert. Die Informationen zu Benutzern werden jeweils bei der Authentisierung vor der ersten Autorisierung durchgeführt. Damit ist sichergestellt, dass die Autorisierung jeweils auf aktuellen Informationen beruht.

Der Datenbankzugriff ist von Seiten der Primärquelle so weit wie möglich einzuschränken. Das Rollensystem benutzt ein eigenes Benutzerkonto für den Zugriff auf die Daten. Die Zugriffe sind auf die nötigen Attribute begrenzt. Das Passwort ist so gewählt, dass es nicht erraten werden kann. Es ist in einer nur für das Rollensystem lesbaren Konfigurationsdatei auf dem Server des Rollensystems gespeichert. Die Verwendung eines Datenbankproxyservers verhindert, dass die Netze miteinander verbunden sein müssen. Ferner kann der Proxyserver Aufgaben eines Application-Level-Firewall-Systems übernehmen und z.B. die Last durch Datenbankzugriffe limitieren.

Damit die Daten der Primärquellen genutzt werden können, nutzt das Rollensystem ein Meta-Schema, in dem die Quelle jedes Attributs sowie die Abbildungsvorschrift für die Schlüsselfelder enthalten sind. Der Zugriff jeder Primärquelle ist über eine Adapterklasse gekapselt, so dass Änderungen in der Datenbank sich lediglich auf die Adapterklasse auswirken. Über eine Brücke kann ferner die Quelle von Attributen entkoppelt werden, so dass es leicht möglich ist, benötigte Attribute aus anderen Datenquellen zu beziehen, falls dies gefordert ist.

5.3 Organisatorische Maßnahmen

Die Gestaltung von IT-Systemen im professionellen Umfeld ist stets mit der Gestaltung von Arbeitsplätzen [20] verbunden. Die Motivation für die Einführung oder Veränderung von IT-Systemen am Arbeitsplatz hängt immer mit der Gestaltung von Prozessen oder der Verbesserung der Arbeitsqualität zusammen. Das trifft insbesondere auf eine umfassende Maßnahme, wie die Gestaltung eines Autorisierungsmanagements zu. Der Hebel einer prozessverändernden Maßnahme setzt bei der organisatorischen Gestaltung an. Die eingesetzte Technik stellt dabei die Mittel zur Verfügung, wohingegen die architekturellen Maßnahmen das Bindeglied zwischen Organisation und Technik darstellen.

Während die Beschreibung der Technik vornehmlich zum Nachweis des Praxisbezugs der Arbeit dient, sich jedoch schnell ändert und stark von den lokalen Gegebenheiten abhängig ist, sind architekturelle Grundlagen sehr viel stabiler und in verschiedene Umgebungen übertragbar. Die Architektur lässt hierbei viele Spielräume, die organisatorisch genutzt werden können und die technisch geeignet gestaltet werden müssen. Die Arbeitsgestaltung beginnt und endet jeweils bei den organisatorischen Maßnahmen, die technisch umgesetzt werden können. Die Architektur ist ein Bindeglied dazwischen und bietet eine Abstraktionsschicht zur Entkopplung von Organisation und Technik.

Im Folgenden gehe ich auf fünf organisatorische Themenbereiche des umfassenden Autorisierungsmanagements ein:

1. Die Nutzung verschiedener RBAC-Merkmale im TUBIS-System,
2. die dezentrale Verwaltung des zentralen RBAC-Systems,
3. die Umsetzung von mehrseitiger Sicherheit in der Authentisierungs-/Autorisierungsinfrastruktur,
4. die strategische Planung für die Infrastruktur sowie

5. die Verwendung von Mustern (Pattern) im Umfeld der RBAC-Modellbildung.

5.3.1 Nutzung verschiedener RBAC-Merkmale im TUBIS-System

Die folgenden RBAC-Merkmale finden im TUBIS-System Verwendung, um die organisatorischen Strukturen im Zugriffsmodell abzubilden:

Static Separation-of-Duty (SSD): SSD-Mechanismen greifen an der TUB bei der Zuordnung so genannter statischer Rollen oder Standardrollen, die automatisch aus den Tätigkeiten der Personen abgeleitet und aus der Datenbank der Personalverwaltung bezogen werden. Solche statischen Rollen können z.B. "Professor", "wissenschaftlicher Mitarbeiter" o.ä. sein. Werden in der Rollenhierarchie nun Anwendungsrollen an diese statischen Rollen gebunden, kann implizit ausgeschlossen werden, dass eine Person in Besitz von bestimmten Kombinationen von Rollen kommen kann.

Dynamic Separation-of-Duty (DSD): Der DSD-Mechanismus wird in dem an der TUB eingesetzten System konsequent auf der Anwendungsebene umgesetzt. Bei Auswahl einer Anwendung aus dem Portalsystem wird der Benutzer jeweils dazu aufgefordert, die Rolle auszuwählen, in der er das System benutzen möchte. Steht ihm nur eine Rolle zur Verfügung, wird diese automatisch aktiviert.

History and object-based SoD: Solche Mechanismen sind an der TUB mittels Kopplung des Rollensystems mit einer Workflow-Komponente geplant, sind jedoch auch zur Zeit, z.B. bei der Beantragung von Subdomains für Webauftritte implementiert. Während die Leiter einer Organisationseinheit der Universität einen Webauftritt beantragen können, müssen Mitarbeiter der Netzwerkabteilung prüfen, ob die Subdomain zur Verfügung gestellt werden kann. Mit Einrichtung des Webauftritts wird die Vergabe der Rechte auf dem neuen Webauftritt allerdings wieder automatisch an die Organisationseinheit gegeben. Hier zeigt sich deutlich die Verteilung der Verantwortlichkeiten, die in diesem Prozess anders abgebildet ist, als in anderen Systemen, wo der Administrator für Einrichtung und Rechtevergabe z.B. universalverantwortlich ist.

Generalisierung, Aggregation und Supervision: An der TU Berlin werden alle drei Prinzipien implementiert. Als Basis dient das Kostenstellenverzeichnis (KST-Verzeichnis) der Organisation. Dieses definiert Einheiten und weist Personen eindeutig ein oder mehreren Kostenstellen zu. Es handelte sich hierbei um eine der vollständigsten Abbildungen der Universität, weshalb sie als Grundlage geeignet ist[1]. Die Rollenzuordnung findet über ein Aggregations-Modell statt. Einer Einheit aus dem KST-Verzeichnis werden eine Menge von Rollen gemäß ihrer Verantwortlichkeit zugewiesen. Die Einheit verteilt dann die Rechte mittels Delegation an die ausführenden Personen. Sobald eine Person Mitglied in einer Rolle wird oder die Mitgliedschaft entzogen bekommt, wird der Person die Änderung ihrer Aufgabe (denn mit der Zuweisung der Rechte sind jeweils Verpflichtungen zur Ausführung verbunden) mitgeteilt.

Neben der Delegation gibt es auch die Vertretung [6]. Hierbei handelt es sich um eine temporäre, nicht übertragbare Weitergabe der Rechte z.B. im Urlaubs- oder Krankheitsfall. Inhaltlich kann eine Vertretungsrolle ferner bedeuten, dass mit der Rolle keine

[1] Mittlerweile wird dazu übergegangen, eine Abstraktionsschicht zu implementieren, die eine benutzerfreundlichere Darstellung der Einheiten ermöglicht.

unmittelbaren Verpflichtungen verbunden sind, sondern mittelbar die Bereitschaft im Bedarfsfall.

Bezogen auf die Sicherheitsmaßnahmen gegen eine fehlerhafte Rollendefinition bedeuten diese Techniken, dass eine Organisationseinheit grundsätzlich nur die Rollen und Rechte delegieren kann, die aus dem Pool ihrer eigenen Verantwortlichkeit stammt. Das sind naturgemäß Rechte, für die die Einheit nötige Fachkenntnis besitzt. Vertretungen können von den Ausführern selbst für die jeweils geeignete Aufgabe definiert werden. Die de facto Vertretungen aus der physischen Welt können hierbei unkompliziert in das Rollenmodell übernommen werden.

Schutz vor zu wenig Rechten: Da an der TU Berlin jede Organisationseinheit mit einer geeigneten Menge von Rechten ausgestattet ist, die von der Einheit selbst verwaltet werden kann, muss vor allem sichergestellt werden, dass immer mindestens ein handlungsfähiger Rollenadministrator zur Verfügung steht, der, wenn nötig, Rollen delegieren kann. Einen Schutz vor fehlenden Zuweisungen von Anwendungsrollen gibt es nicht. Im Bedarfsfall kann die fehlende Zuweisung jedoch in Zusammenarbeit mit dem Anwendungsverwalter bzw. dessen Stellvertreter korrigiert werden.

5.3.2 Dezentrale Verwaltung des zentralen RBAC-Systems

Die Erfahrungen im Bereich Autorisierungsmanagement an der TU Berlin im Vergleich mit den prototypischen Entwicklungen im B2B-Bereich am EU-Projekt MultiPLECX lassen den Schluss zu, dass eine dezentrale Rollenverwaltung in vielen Anwendungsgebieten die geeignete Grundlage für den effektiven Einsatz einer organisationsweiten oder organisationsübergreifenden Zugriffskontrolle ist. Die Delegation der Rollenzuweisung an Personen in den Organisationseinheiten sind der Schlüssel für die Reduzierung der Komplexität in der Rollenverwaltung. Ferraiolo [24] stellt im Kapitel "Role Engineering" verschiedene Organisationen vor, die zwischen 14500 und 186000 Benutzer und 100 bis 3800 Rollen verwalten. Jeder Benutzer besitzt in dieser Aufstellung 1 bis 10 Rollen. Die Gesamtzahl der Rollen in einem dezentral organisierten Rollensystem ist tatsächlich nicht maßgeblich entscheidend. Der Verwalter einer typischen Organisationseinheit an der TU Berlin verwaltet etwa 15 Rollen für 30 bis 80 Personen. Jede Person besitzt etwa fünf Rollen, wobei zwei bis drei dieser Rollen so genannte Standardrollen sind, die die Statusgruppe der Person ("Hochschullehrer", "Studierender", etc.) wiedergeben und somit weder vom Rollenverwalter der Organisationseinheit noch vom Benutzer selbst verwaltet werden müssen. Diese Standardrollen sind allein für die Anwendungsbetreiber relevant, die Anwendungen für eine komplette Statusgruppe zugänglich machen wollen. Der Tatsache, dass auch die vergleichsweise kleine Zahl von zu verwaltenden Rollen und Benutzern eine Herausforderung für die Rollenadministratoren darstellt, wird mit der Entwicklung des xRE-Verfahrens (siehe Kapitel 4) Rechnung getragen.

Die AA-Infrastruktur besitzt sowohl zentrale wie auch dezentrale Aspekte. So wird z.B. immer wieder auf die Umsetzung einer zentralen Sicherheitspolitik hingewiesen. Der Motivation für die Etablierung einer dezentralen Verwaltung von RBAC-Systemen folgt eine Diskussion über dezentrale Verwaltungsmodelle im Allgemeinen und dem in TUBIS implementierten im Speziellen. Dem folgt eine Gegenüberstellung der zentralen und dezentralen Elemente.

Motivation für die dezentrale Verwaltung von RBAC-Systemen

Sandhu beschreibt in [84] die Administration von RBAC-Systemen, kontrolliert durch ein RBAC-System. Die Motivation hinter diesem Vorgehen ist sehr einfach: RBAC-Systeme wurden entwickelt, um Zugriffskontrolle effizienter administrieren zu können. Wird das dafür benötigte Modell so groß, dass dies effizient verwaltet werden muss, so kann hierfür wiederum auf RBAC zur Zugriffskontrolle zurückgegriffen werden. Das RBAC-System kann vereinfacht als Anwendung verstanden werden, die durch ein RBAC-System geschützt wird.

In Kapitel 3.3.2 habe ich bereits weitere Vorteile eines dezentralen Administrationsansatzes vorgestellt. Ergänzend sind ferner die folgenden Argumente hinzuzufügen:

Die adäquate Repräsentation der Organisationsstruktur: Im Fall einer deutschen Universität, wie der TU Berlin kann man von einer verteilten Struktur sprechen. Viele Prozesse sind in Teilen in der zentralen Universitätsverwaltung, in anderen Teilen in den Fakultäten bis hinein in die Sekretariate der Fachgebiete umgesetzt. Mit Hilfe eines dezentral verwalteten Autorisierungssystems lässt sich diese Struktur direkt abbilden.

Die Abbildung der rechtlichen Gegebenheiten: Den Einheiten der Universität wird rechtlich Autonomie in vielen Bereichen zugesprochen. Ein dezentrales RBAC-Modell ermöglicht die technische Implementierung dieser rechtlichen Vorgaben.

Mehrseitige Sicherheit: Ein dezentrales Modell fördert die Verteilung der Verantwortlichkeit im System, erhöht die Transparenz für die Betroffenen und ermöglicht die Verbesserung der Datensparsamkeit an den jeweiligen Administrationspunkten (Zugriff nur auf die jeweils lokal benötigten Daten). Ein zentraler "Superuser" mit universellen Rechten ist nicht mehr notwendig.

Modelle für dezentrale Verwaltung bei RBAC allgemein und in TUBIS

Bei der rollenbasierten Verwaltung von RBAC-Modellen unterscheidet man zwischen administrativen Rollen und Benutzerrollen. Sandhu et al. entwickeln in [84] ein Modell, das zwei getrennte Rollen-Graphen vorsieht: Eine Rollenhierarchie für Benutzerrollen und eine für administrative Rollen. Beide Graphen sind über eine gemeinsame Benutzerbasis miteinander verbunden. Dem gegenüber steht das Crampton-Loizou Modell [17], dass keinen gesonderten Rollengraphen aufbaut. Stattdessen wird der Gültigkeitsbereich für administrative Rollen definiert.

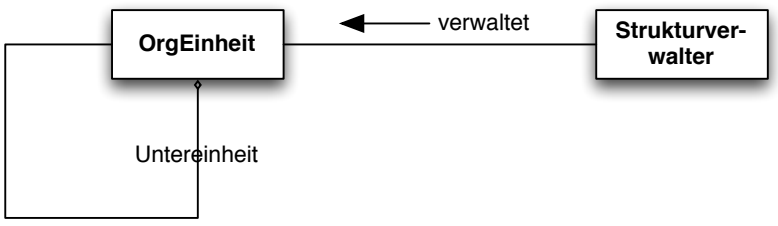

Abbildung 5.14: Klassendiagramm: Strukturverwalter von Organisationseinheiten

Das TUBIS-Modell entspricht einem vereinfachten Crampton-Loizou Modell mit statischen Gültigkeitsbereichen. Die Beziehung des Strukturverwalters für eine Organisations-

einheit kann durch das Klassendiagramm in Abb. 5.14 dargestellt werden. Dabei gilt die Einschränkung, dass nur die enthaltenen Untereinheiten verwaltet werden dürfen, für die es selbst keinen Verwalter gibt. Dieser Sachverhalt kann ergänzend zum Klassendiagramm mit der folgenden OCL Invariante beschrieben werden.

```
context Strukturverwalter
inv: self.Untereinheit.verwaltet->isEmpty()
```

Das Ergebnis dieser Invariante kann im Beispiel wie in Abb. 5.15 dargestellt werden.

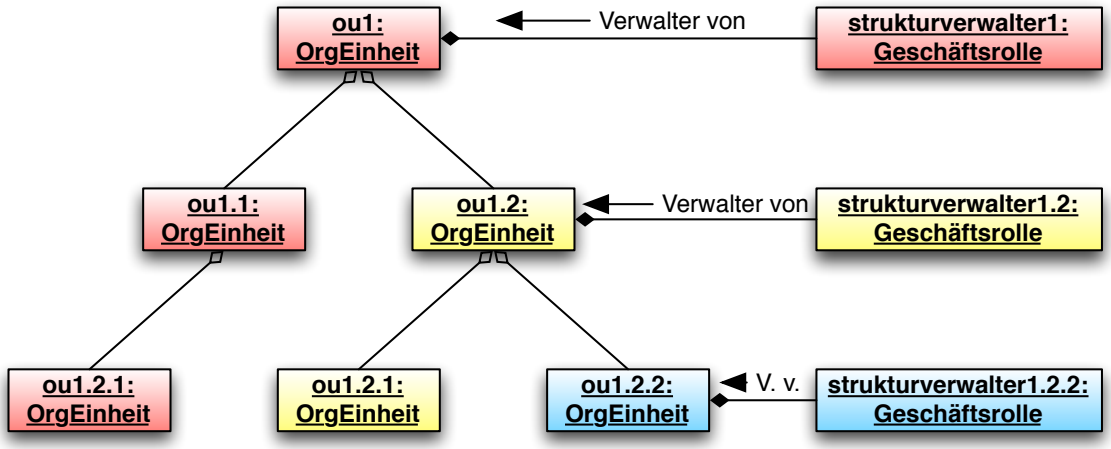

Abbildung 5.15: Objektdiagramm: Strukturverwalter und Rechte auf Organisationseinheiten

Die Vereinfachung des TUBIS-Modells gegenüber dem ARBAC97 und dem SARBAC Modell entsteht durch die Tatsache, dass das TUBIS-Modell die Organisationsstruktur nicht über den Rollengraphen abbildet, sondern die Geschäftsrollen als Teile der Organisationseinheiten modelliert. Ein Strukturverwalter kann immer nur Geschäftsrollen erstellen, ändern oder löschen, die in einer von ihm verwalteten Einheit oder Untereinheit existieren. Die Verwaltung der Struktur selbst ist auf eine andere Sicht ausgelagert.

Diese Modellierung besitzt nicht die Flexibilität von ARBAC97 oder SARBAC, spiegelt jedoch sehr exakt die Befugnisse innerhalb der Universität. Dabei wäre die Verwaltung von Rollenhierarchien innerhalb einer Struktureinheit möglich. Diese Funktionalität wurde bislang jedoch nicht implementiert, weil die relativ geringe Zahl an Rollen und Personen eine flache Organisation praktikabel machen und auf der anderen Seite die Komplexität für die Administratoren verringern.

Typisierung der Rollen

Wie fein die Typisierung gewählt wird, ist abhängig von den involvierten Parteien und der jeweiligen fachspezifischen Sprache. So fällt auf, dass das Stanford Modell sehr viel mehr Rollentypen definiert als das TUBIS Modell. Bei der Einführung des umfassenden Autorisierungsmanagaments an der TU Berlin stellten wir fest, dass eine zu feine Typisierung schnell

zu einer Überforderung führt. So ist das TUBIS Modell im Detaillierungsgrad eher mit dem Modell aus [77] vergleichbar.

Zentrale und dezentrale Elemente

Auch bei verteilt verwalteten Rollenmodellen gibt es Komponenten, die zentral administriert werden, weil bei diesen Elementen eine Verteilung keinen Nutzen bringt oder diese zur Durchsetzung der globalen Sicherheitspolitik zentral geführt werden müssen.

Tabelle 5.1: Aufstellung zentral und dezentral verwalteter Elemente im Rollenmodell

zentrale Elemente	dezentrale Elemente
Modellsichten	Selbstverwaltung (eigene)
Basismodell	Strukturverwaltung
- abstraktes Modell	- Geschäftsrollen
- Vorlagen	- Rollenmitgliedschaften
- Initialisierung	- Identitätsverwaltung
- Automatisierung	- Anwendungsverwaltung
- Organigramm	
Nebenbedingungen	

Die Definition und Bereitstellung der Sichten auf das Modell muss nicht zwingend zentral organisiert werden. Allerdings ist die Sichtbarkeit von Objekten im Modell ein wesentlicher Bestandteil der Sicherheitspolitik. Die Wahl des Basismodells könnte ebenfalls dezentral getroffen werden. Dies würde die Zusammenarbeit der einzelnen Einheiten jedoch sehr viel komplexer gestalten, wäre aber gemäß der Ansätze für verteilte ADF, wie in [45] beschrieben, möglich. Die zentrale Verwendung eines gemeinsamen abstrakten Modells vereinfacht die Prozessbearbeitung innerhalb der Organisation. Zentral gestaltete Vorlagen für Geschäftsrollen sollen die dezentral agierenden Administratoren unterstützen. Der Initialzustand des Modells wird aus Stammdaten und der Organisationsstruktur abgeleitet. Dies geschieht zentral, kann dann aber dezentral geeignet geändert bzw. verfeinert werden. Die automatische Zuweisung von Standardrollen an Hand von Stammdatenattributen aus der Verwaltung hat sich als außerordentlich effizient erwiesen. Standardrollen können in ihrer Form dann genutzt werden, wenn diese auch organisationsweit nach den gleichen Regeln gebildet werden, da sonst die Einhaltung der Sicherheitspolitik nicht gewährleistet werden kann. Die Abbildung der Organisationsstruktur muss zentral erfolgen. Auch bei dezentralen Modellen muss an einer Stelle die Kopplung zentral definiert sein. Über die Definition der Nebenbedingungen werden Regeln der zentralen Sicherheitspolitik im Modell verankert. Auch hier könnten zusätzliche Nebenbedingungen dezentral hinzugefügt werden. Allerdings ist deren Implementierung komplex. Ein Zugang wäre die Definition von speziellen Nebenbedingungen, wie z.B. die in [24, Chapter 5] beschriebenen "Temporal constraints in RBAC".

Demgegenüber steht die dezentrale Verwaltung der eigenen Rollen im Sinne der Vergabe von Vertretungen, die selbstverständlich nur dann genutzt werden kann, wenn dies organisatorisch gewünscht ist. Evtl. müssen die Selbstverwaltungsfunktionen auch in Workflows eingebettet werden und z.B. vom Vorgesetzten bestätigt werden. Innerhalb einer Organisationseinheit können Rollen erstellt, geändert, gelöscht und Rollenmitgliedschaften verwaltet werden. Die Verwaltung von Identitäten geschieht bei den jeweiligen Stellen (z.B. Personal-

stelle und Studierendenbüro), Gäste können jedoch auch dezentral in das System eingetragen werden. Ferner kann es, wie an der TU Berlin nötig sein, weitere Benutzer verteilt in eine zentrale Datenbank einpflegen zu lassen. Jeder Betreiber einer Anwendung innerhalb der AAI kann in die Lage versetzt werden, das Teilmodell seiner Anwendung selbst zu verwalten.

5.3.3 Mehrseitige Sicherheit in TUBIS

Die aktuelle Implementierung des Autorisierungsmanagementsystems an der TU Berlin beachtet nicht nur den Schutz der Ressourcen vor unerlaubtem Zugriff, sondern beachtet ferner die Gefährdungen, die durch Missbrauch, Fehler oder Angriffen von Zulieferern, Herstellern etc. innerhalb der Verwaltung, dem Rechenzentrum oder den Fakultäten entstehen.

Insbesondere folgende Maßnahmen sind im Zusammenhang mit der TUBIS-Realisierung zu nennen:

Verteilung der Informationen: Bei der Implementierung des Identitäts- und Autorisierungsmanagments wird versucht, so weit wie möglich auf zentrale Datenhaltung zu verzichten. Daten werden bei Bedarf aus den Datenquellen angefordert. Neben den Vorteilen (Synchronisation und Aktualität der Daten), die die Vermeidung von Redundanz an dieser Stelle bringt, liegt der Sicherheitsvorteil in der einfachen Tatsache, dass nur die am jeweiligen Ort gespeicherten Daten geschützt werden müssen.

Ein Beispiel hierfür sind die Privatadressen von Studierenden, die früher in der Benutzerverwaltung erfasst wurden. Im Rahmen der Novellierung der Benutzerverwaltung wurde der Bedarf für die Privatadressen neu bewertet. Da die Adressen nur in Fällen von Missbrauch oder bei schwerwiegenden technischen Fehlern benötigt werden, wurde ein Verfahren implementiert, dass die Anforderung der Daten im Bedarfsfall von der Studierendenverwaltung ermöglicht.

Verteilung der Verantwortlichkeiten: Die Existenz eines Superuser-Kontos, also eines Benutzers mit allen vom System her möglichen Rechten ist zur Zeit noch Stand der Technik. Ebenfalls bekannt ist jedoch die Tatsache, dass gerade dieses Benutzerkonto die größte Schwachstelle des Systems darstellt. Die Benutzeroberflächen von TUBIS stellen jeweils eingeschränkte Funktionen zur Verfügung. Die Verteilung der Dienste auf verschiedene Systeme bietet auch die Möglichkeit, den Zugriff selbst für Systemadministratoren einzuschränken. Moderne Betriebssystemerweiterungen stellen ferner Möglichkeiten für die Verteilung von Verantwortlichkeiten zur Verfügung (z.B. Linux [73, 74], Solaris [78]).

Transparenz: Transparenz ist ein wesentlicher Bestandteil für die informationelle Selbstbestimmung der beteiligten Parteien und trägt ferner zum Bewusstsein in Bezug auf die verarbeiteten Informationen bei. Der Mensch stellt ein wesentliches Angriffsziel dar, das durch Social Engineering Angriffe gefährdet ist. Schulungen und Transparenz sind die wichtigsten Maßnahmen gegen solche Angriffe.

TUBIS bietet Transparenz in Form von Selbstbedienungsfunktionen und der Verteilung der Administration auf kleinere, überschaubare Gruppen. Das personalisierte Portal wird auch für die Bekanntgabe von Informationen zu den angebotenen IT-Diensten verwendet. Auch sind an die Nutzung sensibler Dienste Schulungen gekoppelt.

Es gibt ferner Bestrebungen, die Möglichkeiten in der Selbstbedienung in Bezug auf die Kontrolle der personenbezogenen Daten zu verbessern und die Transparenz weiter zu erhöhen.

Pseudonymität und Nichtverfolgbarkeit: Ein Dienstbetreiber muss jeweils die scheinbar gegensätzlichen Anforderungen der Kontrolle über den Dienst und die Datensparsamkeit in Einklang bringen. Im Bedarfsfall muss eine Nutzeraktion zweifelsfrei nachgewiesen werden und es muss nachvollziehbar sein, welche Person die Transaktion wann und mit welcher Berechtigung ausgeführt hat. Auf der anderen Seite sollen die Datenspuren minimiert werden.

An der TU Berlin wird diesen Anforderungen Rechnung getragen, indem Protokolldaten bewusst getrennt gehalten werden, so dass im Bedarfsfall eine Rekonstruktion der Vorgänge durch Zusammenarbeit mehrerer Administratoren möglich ist. Die Aufbewahrung der Protokolle ist jeweils in Absprache mit dem Datenschutzbeauftragten zeitlich begrenzt. Ferner sind die IT Betreiber bestrebt, ein Minimum an Informationen auszutauschen und wo möglich auch pseudonyme Zugänge zu ermöglichen. Diese Funktionalität kann über die AAI bereitgestellt werden.

5.3.4 Strategische Planung für die Infrastruktur

Ein umfassendes Autorisierungsmanagement muss auf allen Arbeitsebenen umgesetzt werden. Das technische Personal muss die nötige Infrastruktur zur Verfügung stellen, sie warten, Fehler beheben und die Benutzer unterstützen. Systemintegratoren müssen die Anwendungen in das System einfügen. Entwickler neuer Anwendungen müssen die Integration berücksichtigen. Die finanziellen und personellen Mittel für den Betrieb eines solchen Systems müssen bereit gestellt werden. Anwendungsbetreiber, Benutzer und Leitung müssen die Integration intendieren, da jede der Gruppen den Erfolg des Systems unterminieren kann.

Bei der Anschaffung neuer Produkte muss die Integrierbarkeit in das Autorisierungsmanagement in die Entscheidungsfindung einfließen. Das ist insbesondere vor dem Hintergrund bemerkenswert, da einige Anbieter versuchen, ihre Produktlinien so aufeinander abzustimmen, dass eine Interoperabilität nur unter den eigenen Produkten möglich ist.

Das Autorisierungsmanagement muss nicht allein die Geschäftsprozesse abbilden, es müssen auch Geschäftsprozesse als Voraussetzung für das Autorisierungsmanagement geschaffen werden. Hierzu gehören Prozesse der Identitäts- und Rollenverwaltung. Die Nutzung des Systems im Sinne der Zugriffskontrolle muss selbstverständlich sein. Die Modellierung über Rollen muss gefördert werden. Demnach ist eine Rückkopplung zwischen Rollensystem und Geschäftsprozessen zu etablieren.

Nutzen die IT-Systeme gemeinsame Daten, auf deren Basis u.a. Zugriffsentscheidungen getroffen werden, ist eine Zusammenarbeit der verantwortlichen Stellen in Hinblick auf die Qualität der Daten unabdingbar. Der Finanzabteilung muss klar sein, dass auf Basis der Kostenstellendefinition Rollen, IT-Strukturen und Ressourcen beeinflusst werden. Die Änderung von Personaldaten hat signifikanten und unmittelbaren Einfluss auf die Berechtigung einer Person. Ferner sind Seiteneffekte zu beachten, wie z.B. die Weiterführung einer lebenslangen Kennung trotz z.B. wechselnder Verträge. Erfassungsfehler oder Änderungen müssen zeitnah korrigiert oder erfasst werden. Die Stellen müssen miteinander reibungslos kooperieren. An der TU Berlin war zu erkennen, dass die Einführung des Systems die Kooperation gefördert hat und im Sinne eines gemeinsamen, konsistenten Datenbestands interagiert wurde.

5.3.5 Muster im TUBIS System

Im TUBIS-Kontext kommt die Verwendung von Mustern hauptsächlich an drei Stellen zum Tragen: Zum einen wurden Muster beim Entwurf des AAI-Rahmenwerkes eingesetzt. Des weiteren werden Backup-Rollen und die rollenbasierte Administration von RBAC-Systemen eingesetzt. Auch ein Delegationsmuster ist für die Vertretungsrollen im Einsatz. Zu guter Letzt werden Muster auch im Sinne von Schablonen eingesetzt. Hierbei geht es um typische Belegungen von Rechtevergaben, die vom lokalen Administrator genutzt und erweitert werden können.

Im Rahmen des eXtreme Role-Engineerings ist der Einsatz von Mustern vorgesehen. Hierbei handelt es sich um Zusammenstellungen von Rollen und Rechten, die für typische Aufgaben jeweils auf eine Organisationseinheit angewendet werden. So gibt es sowohl in Fachgebieten, wie auch in Verwaltungseinheiten an der Universität typischerweise eine oder mehrere Personen, die mit dem Erwerb von Soft- und Hardware betraut ist/sind. Eine geeignete Kombination aus Rollen liegt in diesem Fall als Template vor. Wird dieses Template in einer Einheit angewendet, so werden die Rollen bezogen auf diese Einheit vergeben (also z.B. Besteller für ein bestimmtes Fachgebiet und die entsprechenden Konten). Diese Templates werden von Rollenexperten aus Statistiken abgeleitet, die das XRE-Verfahren erzeugt. Werden also von unterschiedlichen Organisationseinheiten immer wieder sehr ähnliche Kombinationen von Rechten zusammengestellt, so prüfen die Experten an Hand dieser Vorgaben, wie ein geeignetes Template hierfür aussehen würde und stellen dies zur Verfügung. Der im XRE eingesetzte Rollenfindungsdruide versucht an Hand der Vorgaben durch Testfälle, die vom Rollenadministrator der Einheit vorgegeben sind, immer auch Templates zu finden, die möglichst nahe an den Vorstellungen liegen. Die spezialisierten Rollen können dann von den Templates abgeleitet und vom Rollenadministrator der Einheit geeignet erweitert werden.

Selbstverständlich würden mindestens in der Statistik auch Fehlkonfigurationen auffallen, wenn sie häufiger vorkommen. Das Template-Review der Rollenexperten verhindert so zumindest häufig gemachte Fehler. Diese können dann in Folge über Nebenbedingungen verhindert werden und es könnte ein sinnvoller Gegenvorschlag über Templates zur Verfügung gestellt werden.

Wie bei praktisch allen sicherheitsrelevanten Methoden ist die Dokumentation, die Schulung und der Support durch die Experten von größter Wichtigkeit. Eine, wenn nicht *die* entscheidende Angriffsfläche eines Sicherheitssystems ist der Mensch und dessen Beeinflussbarkeit. Wie jedes Sicherheitssystem, ist auch RBAC gegen Social-Engineering anfällig [87]. Dem kann nur mit einem möglichst hohen Wissensstand (Sensibilisierung) der Mitarbeiter begegnet werden. Dazu gehört selbstverständlich auch eine geeignete Benutzerführung der verwendeten Software, ein umfassendes Lehr-, Auffrischungs- und Weiterbildungsangebot, sowie ein kompetenter Support, der im Zweifelsfall Auskunft geben kann.

5.4 Der Anwendungsfall TU Berlin

Der letzte Abschnitt dieses Kapitels fasst konkrete Erfahrungen mit den Benutzern im weitesten Sinne zusammen. Wie wird das System angenommen? Welche konkreten Hürden gab es bei der Einführung und worauf muss auch heute noch im Betrieb geachtet werden? Es handelt sich um Beobachtungen, die es noch zu systematisieren und zu untersuchen gilt. Insofern stellt dieser Abschnitt die Überleitung zum Kapitel "Ausblick" dar.

5.4.1 Das umfassende Autorisierungsmanagement im Universitätsalltag

Die Resonanz zum an der TU Berlin im Einsatz befindlichen umfassenden Autorisierungssystem wurde noch nicht systematisch, z.B. über Fragebögen oder Interviews, erfasst. Jedoch lassen sich einige Aussagen und Trends über Anrufe und Gespräche sowohl bei Projekten, wie auch beim 1st und 2nd Level Support einfangen:

Endnutzer: Viele Endnutzer, vornehmlich aus dem Bereich Lehre, erklären selbst im Fall von Problemen, die der Grund für ihre Kontaktaufnahme sind, dass sie sehr zufrieden bis begeistert vom Portal und der Steuerung über das Rollensystem sind. Zum einen hilft die Zentralisierung der Dienste auf einer Webseite beim Auffinden und Nutzen von Diensten, zum anderen macht sie die Selbstverwaltung im Rollensystem unabhängig von den Administratoren und gibt ihnen nützliche Freiheiten in der Arbeitsorganisation.

Kritik zeigte sich hauptsächlich bei Sonderfällen, die noch nicht oder bislang nur unzureichend berücksichtigt wurden. Lösungen, die mehr als 90% der Fälle abdecken, werden am Telefon gerne auch als "realitätsfremd" beschimpft. Im Sinne der Kundenzufriedenheit ist in solchen Fällen sorgsam abzuwägen, welcher Aufwand für die Implementierung dieser Sonderfälle geleistet werden kann.

Auf der anderen Seite gibt es auch Benutzer, die bestimmte Erwartungshaltungen und Vorstellungen in Bezug auf die Implementierung von Diensten haben. Diese sind unzufrieden, weil die Lösung von ihrer Vorstellung abweicht. An dieser Stelle kann nur wenig für die Kundenzufriedenheit getan werden.

Anwendungsbetreiber: Die Anbieter von Diensten heben in erster Linie die Entlastung in Bezug auf die Kontenpflege hervor. Zwar gibt es noch viele Anrufe und E-Mails von Benutzern in Bezug auf den Zugang, die die Betreiber erreichen. Doch können diese in den allermeisten Fällen an den Kundendienst des Rechenzentrums weitergeleitet werden.

Dem gegenüber steht oft ein meist undifferenziertes "ungutes Gefühl" durch den entstehenden Kontrollverlust. Die Verwaltung der Benutzer spielt sich allein auf der Ebene von Standardrollenzuweisungen oder der Verwaltung von Organisationseinheiten ab. Viele Entscheidungen können die Benutzer selbst treffen. Konkrete Nachteile, die den Anwendungsbetreibern dadurch entstehen oder Probleme in Bezug auf die Anwendungen konnten jedoch bislang nicht benannt werden.

Es gibt jedoch auch Anwendungsbetreiber, die durch die Anbindung an die AAI überfordert sind. Sie treffen auf Supportfälle, bei deren Fehleranalyse mehr als ihr eigenes System betrachtet werden muss. Sie sind in Bezug auf die Fehlersuche im Team nicht trainiert und einigen Administratoren ist es unangenehm, auf Komponenten anderer Administratoren oder auf deren Expertise angewiesen zu sein. Selbst die Rollen- und Rechtedefinition in Zusammenarbeit mit den Portalbetreibern stellt einige Betreiber vor größere Herausforderungen.

An dieser Stelle könnte untersucht werden, wie durch geeignete Werkzeuge und Methoden die Anwendungsbetreiber mehr unterstützt werden könnten.

Kundendienst: Der Kundendienst steht vor der Aufgabe, dem Kunden weiterzuhelfen, obwohl ihm nicht alle Informationen des Kunden zur Verfügung stehen. An dieser Stelle sorgt der umgesetzte Datenschutz für Herausforderungen, die oft und gerne den Ruf nach

mehr Dateneinsicht laut machen. Auch hier werden Werkzeuge erstellt, die es ermöglichen, die für die Unterstützung des Kunden notwendigen Informationen zur Verfügung zu stellen, ohne dabei das Prinzip der Datensparsamkeit zu verletzen.

Die einheitlichen Schnittstellen und das zentralisierte Modell helfen auf der anderen Seite dem Kundendienst im Sinne der Vereinheitlichung von Anwendungsfällen. Auf der einen Seite spielen viele Komponenten im System eine Rolle, auf der anderen Seite sind es für alle Anwendungen die selben Komponenten.

5.4.2 Beispiele für Synergieeffekte

Ein Argument für die Einführung zentralisierter umfassender Dienste sind stets auch erwartete Synergieeffekte. Im Folgenden werden einige Beispiele für erkennbare Synergien an der TU Berlin zusammengestellt:

Die Zusammenführung von Diensten und Daten eröffnet implizit immer auch die Möglichkeit, Verzeichnisse über diese zu erstellen. So bietet das Rollensystem die Möglichkeit, alle zur Verfügung stehenden Anwendungen aufzulisten. Der Benutzer kann so zum einen sehen, welche Anwendungen ihm zur Zeit zur Verfügung stehen aber auch, welche Anwendungen überhaupt existieren. Das bietet ihm die Möglichkeit z.B. mit seinem Strukturverwalter zu erörtern, dass der Zugriff auf weitere Anwendungen für die auszuführenden Tätigkeiten hilfreich wäre.

Die Rollenmitgliedschaften in speziellen Rollen geben Auskunft über mögliche Ansprechpartner in den jeweiligen Einheiten. So kann sich jeder Benutzer anzeigen lassen, wer die verantwortlichen Strukturverwalter, EDV-Koordinatoren oder Leiter bestimmter Einrichtungen sind, um mit diesen in Kontakt zu treten. Die Informationen müssen nicht gesondert gespeichert werden, sondern ergeben sich aus den Rollenmitgliedschaften.

Der Leiter einer Einrichtung kann sich über die Rollenverwaltung eine Liste aller seiner Mitarbeiter erstellen lassen. Das klingt trivial. Es ist jedoch so, dass es vor Einführung des Systems nicht einfach möglich war, zu kontrollieren, ob die Zuordnung in der Personalabteilung tatsächlich dem erwarteten Zustand entsprach. Ferner fließen nun auch externe Mitarbeiter ein, so dass ein vollständiger Überblick über die für die Einheit arbeitenden Mitarbeiter möglich ist.

Vertretungsregelungen müssen nicht mehr pro Anwendung vergeben werden, sondern können zentral über das Rollensystem definiert werden. Die Vertretung für eine Anwendung kann so auch Auswirkungen für alle anderen Vertretungen haben, es sei denn eine Aufteilung ist vom Mitarbeiter gewünscht. Vertretungsregelungen können für den Krankheitsfall oder für den Urlaub definiert werden. Selbst wenn ein Mitarbeiter vorab keine Regelung getroffen hat oder durch Kombinationen von Urlaub und Krankheit Situationen entstehen, in denen keine Vertretung zur Verfügung steht, ist es dem Leiter der Einheit oder einem übergeordneten Leiter möglich, eine Vertretung einzusetzen. Das Hinterlegen von Passwörtern etc. ist auf dieser Ebene nicht mehr nötig.

Die Vergabe von Rollen kann auf einem abstrakten Niveau erfolgen. Als erstes Modul im Bereich Beschaffung wurde an der TU Berlin ein Softwareportal integriert. Nachdem die Rollenmitgliedschaften für die für Beschaffungen zuständigen Mitarbeiter definiert waren, ließ sich das darauf folgende Portal für die Beschaffung von Hardware leicht derselben Rolle zuweisen. Ein neues Ermitteln der Befugten war nicht mehr nötig. Gleiches trifft auf Dienste in Bezug auf Server- oder IP-Betreuung etc. zu. Diese Dienstegruppe wird oft von den gleichen Personen betreut.

Für die Anwendungsbetreiber zahlt sich die gemeinsame Benutzerbasis aus. So werden für Benutzer z.B. bei Malwarebefall Rechner zentral gesperrt und sie erhalten folglich auf kein Anwendungssystem mehr Zugriff. Vergessene Passwörter, neue TAN-Listen oder defekte Chipkarten werden jeweils von zentralen Stellen bearbeitet. Sie betreffen jeweils alle Anwendungen. Bei Änderungen in der Programmlogik oder der Zugriffspolitik können die Anwendungsbetreiber für alle Anwender zentral die Konfiguration ändern, ohne sich mit Benutzern auseinandersetzen zu müssen, die z.B. Mitglied in verschiedenen Statusgruppen und damit Rollen sind. Zuweisungen lassen sich an einer zentralen Stelle ändern.

Als sehr nützlich erweist sich die Möglichkeit, E-Mails an Mitglieder bestimmter Rollen im Sinne einer Mailingliste verschicken zu können, sowie auch an alle einer Statusgruppe zugeordneten Personen (Mitglieder einer Standardrolle) oder alle Mitglieder einer Untereinheit.

5.4.3 Erfahrungen in Bezug auf "Viewpoints" und Vokabeln

Im Laufe der Arbeit im Bereich der Rollenverwaltung traten immer wieder Missverständnisse auf, die im Zusammenhang der jeweiligen Sichtweisen ("Viewpoints") der Kollegen und der Verwendung von Begriffen ("Vokabeln") entstanden. Dieses Problem ist in der Systementwicklung bekannt. Im Folgenden sind bestimmte, insbesondere im Bereich RBAC auftretende Missverständnisse zusammengefasst. Eine Systematisierung und Zusammenfassung könnte in diesem Bereich zukünftige Projekte in diesem Umfeld unterstützen.

Missverständnisse aus der RBAC-Motivation

RBAC ist aus der Motivation heraus entwickelt worden, dass in der Arbeitswelt bereits ein intuitives Verständnis für den Begriff "Rolle" existiert, über den ein leichter Zugang zur Modellentwicklung erreicht werden kann. Der Rollenbegriff wird in der Arbeitswelt im Gegensatz zu den Anforderungen in der Modellbildung unscharf benutzt. Es ist intuitiv also nicht klar, dass es verschiedene Typen von Rollen gibt, die nicht auf die gleiche Weise modelliert werden können. "Datenbankverwalter für die Externendatenbank" ist ebenso eine Rolle, wie "Vorgesetzter" oder "Leiter des Fachgebiets Xy". Es handelt sich jedoch um Rollen auf unterschiedlichen Abstraktionsniveaus. Ferner existieren in der Arbeitswelt Rollen, die sich ausschließlich aus ihren Pflichten definieren, an die jedoch keine Rechte im IT-System geknüpft sind. In einer AAI werden per se nur Rechte verwaltet und überprüft.

An Stellen, an denen die Realisierung sich vom intuitiven Verständnis unterscheidet, treten Missverständnisse auf. Es fällt schwer hinzunehmen, dass ein Teil der Modellierung sich exakt wie erwartet verhält, andere Teile jedoch nicht. An diesen Stellen ist der Begriff "Rolle" oft eher hinderlich.

Ein weiteres Problem entsteht auf Grund der Tatsache, dass die Definition von Rollen in der Realität oft unscharf ist. So kann die Definition einer Rolle in der Arbeitswelt sehr viele Kontextinformationen umfassen, die im RBAC-Modell nicht abgebildet werden können. Auch die Behandlung von Sonderfällen hat ihr Grenzen, wo der Implementierungsaufwand den Nutzen übersteigt. Dies führt dazu, dass die Erwartungshaltung nicht erfüllt wird oder die Implementierung in Details von Regelungen in der Arbeitswelt abweicht. Auch hier entsteht das Problem, dass etwas gleich heißt, sich aber nicht wie erwartet verhält. Ein Beispiel hierfür sind Zahlungsbefugnisse, die in der Realität durch freien Text eingeschränkt werden, der sich nicht systematisiert prüfen lässt. An diesen Stellen wird lediglich die grundsätzliche Befugnis geprüft und die weiterführende Prüfung der Freitextdefinition einem nachgeordneten Prozess

außerhalb des IT-Systems überlassen.

Der Rollenverwaltungs"dienst"

Unter Universitätsmitgliedern wird der Begriff "Dienst" sehr unterschiedlich verstanden. Während sich Informatiker unter einem Dienst oft nicht mehr vorstellen, als ein Softwaresystem, dass über eine Schnittstelle gesteuert werden kann, geht die Erwartungshaltung bis hin zu einer Hotline, der das vage definierte Anliegen unterbreitet werden kann, welches dann im Sinne des Auftraggebers umgesetzt wird.

Die Bereitstellung einer Selbstverwaltungsfunktion in Form einer Webapplikation ist mit dem Begriff "Rollenverwaltungsdienst" nicht für alle Benutzer vereinbar. Es muss damit gerechnet werden, dass einigen Benutzern die Aufgabe der Rollenverteilung als befremdlich bis hin zu einer Zumutung erscheint. Hier muss sensibel mit den Erwartungshaltungen und Begriffen umgegangen werden.

Unschärfe bei der Rollendefinition

Wie bereits erwähnt, werden Rollen in der physischen Welt in der Regel unscharf definiert. Administrative Fehler werden daher mit einer abweichenden Intention entschuldigt ("Das habe ich so nicht gemeint."). Die Rollendefinition im RBAC-System hingegen ist deterministisch. Sie lassen eine Auslegung nicht zu. Es muss daher sorgfältig unterschieden werden, welche Kritik an der Rollendefinition bzw. der Benutzerschnittstellen tatsächlich zu verbessende Aspekte sind und welche Beschwerden alternativ zur Entschuldigung genutzt werden.

Generalisierung in der Informatik

Die Generalisierung ist in der objektorientierten Programmierung ein gängiges Mittel zur Vereinfachung, d.h. zur Reduzierung der Komplexität, zur Wiederverwendung, d.h. Reduzierung des Implementierungs-, Test- und Wartungsaufwands. In der Verwaltung werden Organisationseinheiten, Rollen und Ressourcen zum Teil an Hand anderer Eigenschaften (Attribute und Methoden) unterschieden, als dass es in der Umsetzung im Modell der Fall ist. So kann es vorkommen, dass ein Verwaltungsmitarbeiter die Implementierung als fehlerhaft ansieht, weil nicht die korrekten Begriffe benutzt werden (Ein "Aninstitut" ist kein "Referat", auch wenn diese sich aus Implementierungssicht nicht unterscheiden.). Auch werden in der Verwaltung auf der anderen Seite Objekte gleich klassifiziert, obwohl sie sich aus Sicht der Modellierung signifikant unterscheiden. Es handelt sich dann um eine Ausnahme, die im RBAC-Modell jedoch als eigene Unterklasse oder gar als eigenständige Klasse geführt werden muss.

Um die Irritationen aufzulösen ist es erforderlich, den jeweiligen Objekten sowohl die IT-technische, wie auch die verwaltungstechnische Klassifizierung mitzugeben und entsprechend zu verwenden.

Unterscheidung von Anwendungs- und Geschäftsrollen

Aus den Erfahrungen an der TU Berlin kann geschlossen werden, dass die Unterscheidung von Anwendungs- und Geschäftsrollen den Benutzern Schwierigkeiten bereitet. Der Schritt "Frau X soll Anwendung A benutzen können.", ist leicht nachvollziehbar. Die Abstraktion "Frau X ist eine G und alle G können die Anwendung A benutzten", ist zumeist eine Indirektion zu

viel. Dabei zahlt sich die Indirektion bereits bei der Definition von Vertreterschaften aus. Wie aber sollen die korrekten Geschäftsrollen definiert werden?

Zur Auflösung dieses Dilemmas wurde das xRE Verfahren entwickelt. Es soll helfen, an Hand von Personen-Anwendungs-Zuordnungen die richtigen Indirektionsschritte zu finden.

Kapitel 6

Ausblick

> Lt. Ford: Is time-travel possible? Dr. McKay: Well, according to Einstein's General Theory of Relativity, there's nothing in the laws of physics to prevent it. Extremely difficult to achieve, mind you, you need the technology to manipulate black holes to create wormholes not only through points in space, but time.
> Maj. Sheppard: Not to mention a really nice De Lorean.
> Dr. McKay: Don't even get me started on that movie!
> Maj. Sheppard: I liked that movie!
>
> Stargate Atlantis (TV series)

Das umfassende Autorisierungsmanagement wurde stets als System vorgestellt, das sowohl auf technischer, wie auch auf architektonischer und organisatorischer Ebene umzusetzen ist. Weiterführende Arbeiten können jeweils auf allen drei Ebenen erfolgen. Dabei wird zum einen die Weiterentwicklung des in Produktion befindlichen Systems betrachtet und zum anderen die wissenschaftliche Auseinandersetzung mit der Thematik.

6.1 Weiterführende Analyse des existierenden Systems

Ein praktisches Ziel, das sich dieser Arbeit anschließen wird, ist die Optimierung des Portalangebots an der TU Berlin. Um die Verbesserung wissenschaftlich fundiert vorantreiben zu können, wäre es hilfreich, verlässliche Statistiken zur Nutzung der einzelnen Portalanwendungen zu erstellen und diese ferner mit Befragungen und Beobachtungen von ausgewählten Musternutzern zu vervollständigen. So kann ein konsistentes Bild der Nutzung von Anwendungen entstehen, das dazu genutzt werden kann, die Verbesserung der Angebote an der Nutzung zu orientieren, Schwachstellen in Bezug auf die Benutzungsfreundlichkeit aufzudecken oder Tendenzen abzuleiten, die den zukünftigen Ausbau des Anwendungsangebots steuern könnte.

Um sicherzustellen, dass mit den Portalanwendungen tatsächlich auch eine objektive Verbesserung der betroffenen Arbeitsumfelder verbunden ist, müsste vor und nach Einführung der Portalnutzung eine Arbeitsanalyse (z.B. nach dem KABA-Verfahren [20]) durchgeführt werden. Die über diese Verfahren aufgedeckten Schwachstellen könnten dann überarbeitet werden. Eine darauf folgende Arbeitsanalyse wäre dann in der Lage, die erfolgreiche Verbesserung zu dokumentieren. Solche Arbeitsanalysen sind in jedem Fall als eigene, vollständige

Projekte zu sehen. Neben konkreten Verbesserungen im Arbeitsumfeld an der TU könnten aus den Untersuchungen jedoch auch allgemein nützliche Erkenntnisse in Bezug auf die Nutzung von Webportalen oder rollenbasierten Autorisierungssystemen erlangt werden.

6.2 Unterstützung der Anwendungsbetreiber

Der deutliche Schwerpunkt dieser Arbeit liegt auf der Konstruktion einer Authentisierungs- und Autorisierungsinfrastruktur mit Eigenschaften, die in verschiedenen Bereichen, wie z.B. an einer Universität, besondere Vorteile gegenüber anderen Lösungen bieten und der Unterstützung insbesondere der Rollenadministratoren in Untereinheiten der Organisation. Diese Unterstützung wird im Zusammenhang einer globalen Administration und der Entwicklung oder Einbindung von Anwendungen in der Infrastruktur betrachtet. Eine Weiterführung dieser Arbeit könnte darin bestehen, zusätzliche Unterstützungen für den Integrationsprozess bei Anwendungen zu entwickeln. Dieser könnte sowohl eine Systematik beinhalten, wie die unterschiedlichen technischen Integrationsvarianten idealerweise zu nutzen sind als auch ein Verfahren für den Entwurf von Anwendungsrollen, das in das xRE Verfahren greift. Alle vorkommenden Anwendungsfälle für die Anwendungsverwaltung sind im xRE Verfahren abgebildet. Es wäre jedoch hilfreich, gesonderte Methodiken zu entwickeln, wann welche Fälle von Seiten des Anwendungsadministrators zu betrachten sind.

6.3 Verschiedene Dimensionen der Integration

Nicht nur architektonisch und organisatorisch würde sich eine Fortführung der Arbeit in Richtung Anwendungsintegration lohnen. Selbstverständlich wäre auch eine technische Weiterentwicklung für die Integration hilfreich, denn das Anwendungsangebot ist allein an der TU Berlin heute noch lange nicht erschöpft. Verbesserte Techniken für die Anwendungsintegration wären vielen Projekten hilfreich.

Die zur Zeit genutzten Synergieeffekte in Bezug auf E-Mail, sind nur der Anfang einer möglichen Integration von RBAC-Systemen und elektronischer Post. Bereits in [42] sind weiterführende Ansätze diskutiert, die weit über die heutige Integration hinausgehen, indem das RBAC-System nicht nur für die Ermittlung von Mitgliedern und deren E-Mailadressen genutzt wird, sondern auch zur Verschlüsselung bzw. Umverschlüsselung für Mailinglistenmitglieder.

Auch die Aspekte der mehrseitigen Sicherheit lohnt es sich, weiter zu verfolgen und die Arbeiten [47] und [44, 46] fortzusetzen. Insbesondere wird es Zeit, die verschiedenen Sichtweisen dessen, was unter "Identitätsmanagement" verstanden wird, nämlich zum einen die Verarbeitung von Personendaten zur Identifizierung in Systemen und zum anderen die Verwaltung der eigenen Teilidentitäten mit dem Ziel der Steuerung der über eine Person bekannten Informationen für Dritte zu vereinen. So könnte das RBAC-System so erweitert werden, dass ein Benutzer vor Nutzung einer Anwendung zunächst über die durch die Anwendung benötigten Informationen aufgeklärt wird und dieser dann unterscheiden kann, ob er der Nutzung dieser Daten zustimmt, ein evtl. verkleinertes pseudonymes oder anonymes Angebot nutzen möchte oder auf die automatische Weitergabe seiner Daten verzichtet. Dies ist insbesondere interessant, wenn die Daten wie im Fall der TU Berlin, nicht nur innerhalb der Universität weitergegeben werden, sondern auch an Auftraggeber, also Dritte.

6.4 Organisationsübergreifendes RBAC

In seinem Ausblick zur Entwicklung von RBAC [83] erwähnt Ravi Sandhu auch den Artikel [45] und sieht interessante Möglichkeiten in der Entwicklung von organisationsübergreifenden RBAC-Systemen. Diesen Aspekt hatten wir auch in [30] und [31] im Fokus behalten, jedoch nicht weiter entwickelt. Lösungen zur Kooperation mit Rollen, wie Shibboleth, nutzen heute nur einen Bruchteil der Potentiale, die wir 1999 und 2000 beschrieben hatten. Protokolle zur Kommunikation verschiedener ADF wären eine Schlüsseltechnologie, die auch in Richtung Datenschutz eine Weiterentwicklung darstellen würden. Einen anderen Ansatz jedoch mit ähnlichen Zielsetzungen verfolgt beispielsweise dass Access eGov Projekt [59] (`http://www.accessegov.org/`). Verteilte Zugriffsentscheidungssysteme würden ferner die RBAC-Systeme auch im Haus skalierbar machen und böten wichtige Möglichkeiten zur Erhöhung der Sicherheit der Autorisierungsinfrastruktur durch Verteilung der Verantwortlichkeit und Systeme.

6.5 Lebenszyklus des Autorisierungsmanagements

Die vorliegende Arbeit lässt den Zeitaspekt komplett außer Acht. Eine Anforderung eines umfassenden Autorisierungsmanagements könnte sein, Zugriffe in zeitlicher Abfolge nachvollziehen zu können; d.h. auf Grund welcher Berechtigungskonfiguration hat ein bestimmter Zugriff zu einem gegebenen Zeitpunkt stattgefunden. Damit einher geht auch die Frage der Versionskontrolle im Rollenmodell. Die Versionskontrolle ist auch bei der Softwareentwicklung eine wichtige Schlüsseltechnologie. Beim Festhalten von Rollenkonfigurationen wäre das Festhalten von Konfigurationsständen, das Einspielen von Teiländerungen oder das Rückverfolgen von Konfigurationen zu einem bestimmten Zeitpunkt von hohem Nutzen!

6.6 Vision mehrseitig sicherer AAA-Infrastrukturen

Die in dieser Arbeit beschriebenen Komponenten zur Authentisierung, Autorisierung und zum Auditing wurden bislang zum Teil nur prototypisch entwickelt und sind bislang nicht als Ganzes, also als komplette mehrseitig sicherere AAA-Infrastruktur, umgesetzt worden. Dabei wären z.B. noch die Fragen zu klären, wie die Schnittstellen gestaltet sein müssten, um nicht durch Integration Lücken in das System zu bringen. Die Vervollständigung der hier skizzierten Vision wäre eine mögliche und logische Fortsetzung dieser Arbeit.

6.7 Musterverzeichnis für RBAC-Systeme

Mit "Role-Based Access Control" von Ferraiolo et al. [24] und "Role Engineering for Enterprise Security Management" von Coyne et al. [16] gibt es eine hilfreiche Zusammenstellung des Wissens über RBAC. Beide Bücher orientieren sich jedoch sehr am RBAC-Modell und systematisieren mittels mathematischer Modelle. Wie in dieser Arbeit dargelegt, lassen sich Muster für und in RBAC-Systemen auf unterschiedlichen Ebenen finden und anwenden. Konsequent wäre daher eine Zusammenstellung dieser Muster mit Schwerpunkt auf die Software- und Modellentwicklung, wie in [11] und [28].

6.8 Modellierungssprache(n)

Ein Aspekt, der den Gedanken des RBAC-Systems in Bezug auf die Software-Entwicklung fortführt, ist die Idee einer objektorientierten Hochsprache zur Beschreibung von RBAC-Modellen, im Gegensatz zu den heute üblichen Beschreibungen auf Basis von XML [5, 32]. Diesen Gedanken verfolgten wir in [30]. Eine solche Sprache könnte beliebige Rollenmodelle zulassen, jedoch mit einer Standardbibliothek ausgestattet sein, die die gängigen RBAC-Muster abbilden kann. Vor allem im Hinblick auf die Definition komplexer Nebenbedingungen könnte eine solche Sprache die RBAC-Modellierung weiter bringen. Es wäre ferner möglich, eine gewisse Austauschbarkeit zwischen Modellteilen des gleichen Interpreters zu erreichen. Wie aber müsste ein Interpreter geschaffen sein, um in einer solchen Hochsprache entwickelte Modelle auch mit tausenden von Rollen und vielen tausend Benutzern noch nutzbar zu machen? Bieten hier Persistenzmodelle zum Ablegen von Modellobjekten einen Ansatz? Helfen Interpreter/Compiler-Kombinationen, wie die bereits erwähnten Jython/Java oder Beanshell/Java Kombinationen bei der Entwicklung solcher Systeme weiter? Und wie werden solche Modelle wiederum für die unterschiedlichen Benutzer dargestellt?

6.9 RBAC-Kontrolle bis in die Persistenzschicht

Als mit der Arbeit an TUBIS begonnen wurde, arbeiteten wir an der Vision eines RBAC-Schutzes, der bis in die Speicherung der Modellobjekte reichte. Die Idee war, dass selbst die Administratoren des Servers, auf dem das RBAC-Modell lokalisiert ist, keinen Zugriff auf RBAC-Objekte anderer Benutzer erhielten. Ein solches Konzept wäre z.B. mit Hilfe von RBAC-Modulen im Betriebssystemkern möglich, wie z.B. mittels RSBAC [73, 74]. Ein anderer Ansatz wäre eine Kombination aus Verschlüsselung und Verteilung von Verantwortlichkeiten. Die Trennung kann dann innerhalb des Betriebssystems über so genannte Jails bzw. Partitions oder mit Hilfe von Virtualisierung oder der Trennung von Servern erfolgen. Die Fortsetzung der Arbeit in diesem Bereich hätte vornehmlich die Frage zu klären, wie eine ADF so entworfen werden kann, dass sie verteilt laufen kann und keinen universellen Zugriff auf alle Objekte benötigt. Hier gibt es viele Berührungspunkte mit der Arbeit am organisationsübergreifenden RBAC. Andere Berührungspunkte finden sich in Bezug auf Arbeiten an sicheren Betriebssystemarchitekturen wie z.B. an der TU Dresden [36, 37] (`http://os.inf.tu-dresden.de/`).

Kapitel 7

Schlussbetrachtung

No more training do you require.
Already know you that which you need.

Master Yoda

Ziel dieser Arbeit war der Entwurf und die Evaluation eines Autorisierungsmanagements mit verteilter, dezentraler Administration bei Durchsetzung einer zentralen Sicherheitspolitik. Das Autorisierungssystem sollte geeignet sein, den Zugriff auf die wesentlichen Prozesse einer Organisation oder Föderation durch ihre Mitglieder und Nutzer zu steuern. Dabei wurde nachgewiesen, dass bei der Realisierung mehrseitige Sicherheit Anwendung finden kann und insbesondere Pseudonymität und Nichtverfolgbarkeit anforderungsgemäß umgesetzt werden können. Im Rahmen der Arbeit wurde ein Verfahren für einen benutzerfreundlichen Rollenentwurf entwickelt, der speziell auf die dezentrale Administration zugeschnitten ist und durch die Vermeidung unnötiger Redundanzen und eine transparente Rollendefinition zur Erhöhung der Sicherheit des umfassenden Systems beiträgt.

Rollenbasierte Zugriffskontrolle ist heute in vielen Bereichen Stand der Technik, wenn es um den Schutz digitaler Dienste oder Ressourcen geht. Gerade in großen Organisationen oder bei organisationsübergreifenden Systemen kann RBAC dann sein Potential als effizientes Autorisierungsmodell voll ausspielen, wenn es verteilt administriert eingesetzt wird. Zur Einbettung in bestehende Strukturen und im Sinne der Verteilung von Verantwortlichkeiten aus Sicherheits- oder Mitbestimmungserwägungen eignet sich ein mehrteiliges Modell. Ein solches Modell wurde im Rahmen dieser Arbeit entworfen. Als geeignet hat sich eine Trennung von Organisationsstruktur (Organigramm), Geschäftslogik und Applikationslogik herausgestellt. Auch die Verwaltung von Personendaten, also das Identitätsmanagement, kann als isolierter Teil des Modells interpretiert werden. Naturgemäß sind an zentrale Komponenten einer Sicherheitsinfrastruktur, wie das Autorisierungssystem, höchste Sicherheitsanforderungen zu stellen. Dabei reicht es nicht aus, die Gefährdungen einseitig, also nur im Sinne des Schutzes vor unerlaubter Nutzung von Diensten zu betrachten. Im Sinne der Sicherheit der IT-Infrastruktur der Organisation und im Hinblick auf die Erfüllung rechtlicher Regelungen und Verordnungen ist die Berücksichtigung der Aspekte und Techniken der mehrseitigen Sicherheit unverzichtbar. Es wurde gezeigt, wie Verteilung von Informationen und Verantwortlichkeiten in einem umfassenden Autorisierungsmanagement umsetzbar sind und wie Pseudonymität und Nichtverfolgbarkeit in den Bereichen Authentisierung, Autorisierung und Auditing gewahrt bleiben können. Da die Sicherheit und Effektivität des Autorisierungssystems stark von der Modellierung des Rollenmodells abhängen, muss dem Role-Engineering Prozess besondere Aufmerksamkeit zuteil werden. In dieser Arbeit wurde untersucht, wie dabei Erkenntnisse des Software Engineering auf Rollenmodelle angewendet werden können. So können Muster auch in Rollenmodellen helfen, Expertenwissen für unterschiedliche Benutzergruppen nutzbar

zu machen. Zur Verbesserung der Benutzungsfreundlichkeit beim Role-Engineering wurde das eXtreme Role-Engineering Verfahren entworfen und prototypisch implementiert. Eine Evaluation mit Daten aus einem produktiven System zeigte den praktischen Nutzen des Verfahrens.

Die Implementierung des Autorisierungssystems TUBIS an der Technischen Universität Berlin ist seit nunmehr zwei Jahren im Einsatz. Die Erfahrung zeigt, dass das dreiteilige Modell, die verschiedenen eingesetzten Integrationsmethoden für die Anwendungen, sowie der Einsatz von Rollentemplates als vereinfachte Art von Rollenmustern die definierten Anforderungen erfüllt. Die Rückmeldungen der Benutzer sind in sehr großen Teilen positiv, die Zahl der Anwender und integrierten Anwendungen steigt. Durch die verteilte Administration und die Integration von Anwendungen, die ohne das TUBIS-gesicherte Portal nicht möglich wären, wurden Kapazitäten unter den Administratoren im Rechenzentrum frei, die nun für die Modernisierung der Infrastruktur und die Erweiterung des Dienstleistungsangebots zur Verfügung gestellt werden können.

Da die Anforderungen an das vorgestellte System aus sehr unterschiedlichen Anwendungsszenarien stammt, kann davon ausgegangen werden, dass es weit gestreute Bereiche gibt, in denen die Lösungen eingesetzt werden können. Der Einsatz eines mehrteiligen RBAC-Modells ist dabei nur dort sinnvoll, wo eine Verteilung der Verantwortlichkeiten für die einzelnen Teile erwünscht und organisatorisch umsetzbar ist bzw. wo allein die Größe des Modells dies sinnvoll erscheinen lässt. In kleineren Organisationen kann der Aufwand der Administration schnell gegenüber einer zentralen Lösung steigen. Es sind jedoch auch Anforderungen denkbar, die dennoch den Einsatz eines mehrteiligen Modells rechtfertigen würden, wie die Überlegung zur Verteilung der Verantwortung oder die Überlegungen zur Pseudonymität und Nichtverfolgbarkeit zeigen. Die in dieser Arbeit vorgestellten Methoden zur Umsetzung dieser Merkmale der mehrseitigen Sicherheit sind mit einem vergleichsweise hohen organisatorischen und technischen Aufwand verbunden, so dass abzuwägen ist, welche Methoden zur Erreichung der jeweiligen Schutzziele im entpsrechenden Einsatzgebiet Anwendung finden sollten.

Das eXtreme Role-Engineering Verfahren wurde zur Administration kleiner Organisationen und von Einheiten einer großen Organisation entwickelt. Die Administration von größeren Organisationen wäre ebenfalls denkbar. Es ist jedoch vorab abzuschätzen, wie groß der Refaktorisierungsaufwand bei der Anwendung des Verfahrens wird. Eine pauschale Aussage hierzu lässt sich nicht treffen. Entworfen und evaluiert wurde das Verfahren für Einheiten mit bis zu 50 nicht hierarchischen Rollen und weniger als 100 Mitgliedern pro Einheit. Der Aufwand des Verfahrens ist allein von den Rollen und damit verknüpften Rechten abhängig. Der Refaktorisierungsaufwand ist stark von der Verzahnung der Rollen und Rechte untereinander abhängig. Neben der Zahl der Rollen und der Abhängigkeiten der Rollen untereinander ist auch der Umfang der Änderungen am Modell relevant. Das Verfahren setzt kleine iterative Änderungen voraus. Für jede neu zu definierende Rolle oder die Verknüpfung mit einem neuen Recht muss eine Iteration im Verfahren durchlaufen werden. Bei umfangreichen Änderungen im Rollenmodell bieten sich demnach alternative Role-Engineering Verfahren an.

Die quantitative Bestimmung der Konfigurationen, in denen verschiedene Role-Engineering Verfahren am effizientesten sind, könnten Bestandteil der weiterführenden Forschung auf diesem Gebiet sein. In die gleiche Richtung gehen Untersuchungen zur konkreten arbeitswissenschaftlichen Evaluation von Modell und eXtreme Role-Engineering Verfahren. In Bezug auf das eXtreme Role-Engineering Verfahren wäre ein Ausweitung der möglichen Anwendungsgebiete zu untersuchen. Das gesamte System könnte nicht nur verteilt administriert und modelliert, sondern auch in verteilten Serverumgebungen implementiert werden. In diesem Zusammenhang sind auch Forschungen in Bezug auf organisationsübergreifende RBAC-Systeme zu

sehen.

Theoretische Erwägungen weisen darauf hin, dass die verwendeten Algorithmen dazu neigen, das Rollenmodell zu fragmentieren, also viele Rollen mit wenigen Rechten zu erzeugen. Dieses unerwünschte Verhalten konnte in den prototypischen Tests nicht beobachtet werden. Die Fragestellung sollte jedoch weiter untersucht werden.

Es muss ferner bedacht werden, dass eXtreme Role-Engineering nicht für die Rollenzuteilung geeignet ist und auch keine Methoden zum Löschen von Rollen implementiert. Es ersetzt also keine klassische Rollenverwaltung.

Für weiterführende Entwicklungen ist die Fortsetzung der Arbeiten vor allem in Bezug auf die Anwendung von mehrseitiger Sicherheit auf den Bereich Authentisierung, Autorisierung und Auditing sowie die Systematisierung des Themenkomplexes RBAC in Form von Entwurfsmustern erstrebenswert.

Anhang A

Verzeichnisse

Fachwörterverzeichnis

In dieser Tabelle sind die in der englischsprachigen Fachwelt üblichen Fachbegriffe den von mir gewählten Übersetzungen gegenübergestellt.

Deutsch	Englisch (original)
Anfrage	request
Anwendungsfall	use-case
Anwendungsrollen	application roles
Benutzungsschnittstelle	user interface
Dekorierer-Muster	decorator pattern
Entscheidung	decision
Entscheidungsdurchsetzungsinstanz	access enforcement facility
Föderation	fedaration
Freigabestufe	clearance level
Geschäftsrollen	business roles
Identitätsmanagement	identity management
mehrseitige Sicherheit	multilateral security
Muster	pattern
Nebenbedingung	constraint
Objekt	object
Provisionierung	provisioning
Rahmenwerk	framework
Refaktorisierung	refactoring
Rolle	role
Rollenbasierte Zugriffskontrolle	role-based access control
Rollenentwurf	role-engineering
Sicht	view/viewpoint
Sichtenorientierter Modellierungsansatz	modeling using viewpoints
Subjekt	subject
Verteilung von Verantwortlichkeiten	separation-of-duty
zielorientiert	goal-based
Zugriffsentscheidungseinheit	access decision facility

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Data Networks and Open System Communications Security: Open Systems Interconnection - Security Frameworks for Open Systems: Access Control Framework. ITU Recommendation X.821, November 1995.

[2] IAM Wiki Lexikon. URL `http://www.iam-wiki.org/Lexikon`.

[3] Open source identity management, Januar 2006. URL `http://safehaus.org/map/jan06/`.

[4] Gail-John Ahn and Michael. E. Shin. Role-based authorization constraints specification using object constraint language. In *6th IEEE International Workshop on Enterprise Security (WETICE 2001), MIT, MA*, June 20-22 2001.

[5] A. Anderson. XACML profile for role based access control (RBAC). *OASIS Access Control TC committee draft*, 1:13, 2004.

[6] Ezedin Barka and Ravi Sandhu. Framework for role-based delegation models. Technical report, Laboratory of Information Security Technology, Information and Software Engineering Department George Mason University, Fairfax, VA 22030, USA, 2000.

[7] Jörg Bartholdt, Thomas Hildmann, and Klaus Nagel. Abgesicherte Internet-Umgebungen mit Hilfe rollenbasierter Zugriffsmechanismen für WWW- und Email-Dienste. In *Proceedings of 5. Workshop "Sicherheit in vernetzten Systemen"*. DFN-Cert und DFN-PCA, 1998.

[8] Oliver Belikan. Identity & Access Management: Funktionsbeschreibung eines Identity & Access Managements. Technical report, duebleSlash Net-Business GmbH, Müllerstr. 12B, 88045 Friedrichshafen, Mai 2006.

[9] D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations. Mitre Corporation, November 1973.

[10] R. Buschermöhle, H. Eekhoff, and B. Josko. Erfolgs- und Misserfolgsfaktoren bei der Durchführung von Hard- und Softwareentwicklungsprojekten in Deutschland. BIS-Verlag, Oldenburg, September 2006. URL `http://vsek01.informatik.uni-oldenburg.de/~joomla/content/SUCCESSOnlineVersion28092006.pdf`.

[11] Frank et.al. Buschmann. Pattern-orientierte software-architektur. Addison-Wesley, Bonn, Reading, Massachusetts, ..., 1998.

[12] Ramaswamy Chandramouli. Application of XML tools for enterprise-wide RBAC implementation tasks. In *Proceedings of the fifth ACM workshop on Role-based access control*, pages 11–18, New York, NY, USA, 2000. ACM. ISBN 1-58113-259-X.

[13] L. Chaum, David. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/358549.358563. URL `http://portal.acm.org/ft_gateway.cfm?id=358563&type=pdf&coll=GUIDE&dl=GUIDE&CFID=27567668&CFTOKEN=76565559`.

[14] D. D. Clark and D. R. Wilson. A comparison of commercial and military computer security policies. In *IEEE Symposium of Security and Privacy*, pages 184–194, 1987.

[15] Edward J. Coyne. Role engineering. In *ACM RBAC Workshop*, MD USA, 1996.

[16] Edward J. Coyne and John M. Davis. *Role Engineering for Enterprise Security Management*. Information Security and Privacy Series. Artech House, 2008.

[17] Jason Crampton and George Loizou. Administrative scope: A foundation for role-based administrative models. *ACM Trans. Inf. Syst. Secur.*, 6(2):201–231, 2003. ISSN 1094-9224. doi: http://doi.acm.org/10.1145/762476.762478. URL `http://portal.acm.org/ft_gateway.cfm?id=762478&type=pdf&coll=GUIDE&dl=GUIDE,GUIDE&CFID=27372616&CFTOKEN=19972970`.

[18] Fredj Dridi, Björn Muschall, and Günther Pernul. Administration of an RBAC system. In *Proc. of the 18th IFIP International Information Security Conference (SEC 2003)*, Athens, Greece, Mai 2003. URL `http://ieeexplore.ieee.org/iel5/8934/28293/01265447.pdf`.

[19] G. Dueck. Projekte, Strukturen und Herzblutenergie. *Informatik-Spektrum*, 31(6):613–618, 2008.

[20] H. Dunckel, W. Volpert, M. Zölch, U. Kreutner, C. Pleiss, and K. Hennes. *Kontrastive Aufgabenanalyse im Büro. Der KABA-Leitfaden – Grundlagen und Manual*. Stuttgart, 1993.

[21] J. Eckstein. XP–eXtreme Programming: Ein leichtgewichtiger Software-Entwicklungsprozess. *basicpro*, 35:6–1, 2000.

[22] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. In *ACM Transactions on Information and Systems Security*, volume 4, pages 224–274, August 2001.

[23] David F. Ferraiolo, Janet A. Cugini, and D. Richard Kuhn. Role-based access control (rbac): Features and motivations. In *Proceedings of 11th Annual Computer Security Application ...*, 1995.

[24] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-Based Access Control*. Artec House, second edition edition, 2007.

[25] Anthony Finkelstein, Jeff Kramer, and Michael Goedicke. Viewpoint oriented software development. In *Proc. of Third Int. Workshop on Software Engineering and its Applications*, Toulouse, December 1990.

[26] Independent Centre for Privacy Protection (ICPP) Schleswig-Holstein and Studio Notarile Genghini (SNG). Identity management systems (ims): Identification and comparison study, September 2003.

[27] Gesellschaft für Informatik e.V. (GI). Extremes programmieren (xp), November 2008. URL `http://www.gi-ev.de/no_cache/service/informatiklexikon/informatiklexikon-detailansicht/meldung/extremes-programmieren-xp-40/`.

[28] Erich Gamma. Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software. Bonn, 1996.

[29] Shu Gao, Zhengfan Dai, and Huiqun Yu. Improving scenario-driven role engineering process with aspects. In *Erly Aspects Workshop in Conjunction with the OOPSLA Converence*, Vancouver, Canada, October 24-28 2004.

[30] Thomas Gebhardt and Thomas Hildmann. Enabling technologies for role based online decision engines. In *Fifth ACM Workshop on Role-Based Access Control*, July 2000.

[31] Thomas Gebhardt and Thomas Hildmann. Rollen als Schlüssel für B2B-Anwendungen. *DuD - Datenschutz und Datensicherheit*, 24 (2000) 10, 2000.

[32] S. Godik, A. Anderson, B. Parducci, P. Humenn, and S. Vajjhala. OASIS eXtensible Access Control 2 Markup Language (XACML) 3. Technical report, Technical report, OASIS, 2002.

[33] Pia Grund-Ludwig. Identity Managment - Freies Geleit durch den Passwort-Dschungel. Handelsblatt.com, Oktober 2006.

[34] Klaus Hamann. Entwurf und prototypische Implementierung eines sicheren Logging-Mechanismus. Diplomarbeit, Technische Universität Berlin, Fachbereich 13, Institut für Softwaretechnik, Januar 2001.

[35] Hannes Federrath and Andreas Pfitzmann. *Handbuch IT in der Verwaltung*, chapter IT-Sicherheit, pages 273 – 292. Springer-Verlag Berlin, 2006.

[36] H. Härtig. Security architectures revised. In *10th ACM SIGOPS European Workshop*. ACM, September 22-25 2002.

[37] H. Härtig, M. Hohmuth, N. Feske, C. Helmuth, A. Lackorzynski, F. Mehnert, and M. Peter. The nizza secure-system architecture, proceedings of collaboratecom 2005. *San Jose, CA, USA*, 2005.

[38] Q. He. A goal-driven role engineering process for privacy-aware rbac systems. In *Proc. of the 11th IEEE International Requirements Engineering Conference (RE'03) Doctoral Symposium*, pages 31–35, Monterey Bay, CA, September 8-12 2003.

[39] Q. He. Privacy enforcement with an extended role-based access control model. Technical Report TR-2003-09, NCSU Computer Science, February 2003.

[40] Q. He and A. I. Antón. A framework for modeling privacy requirements in role engineering. In *Proc. of the 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03)*, pages 137–146, Klagenfurt/Velden, Austria, June 16-17 2003.

[41] T. Hildmann and C. Ritter. TUBIS-Integration von Campusdiensten an der Technischen Universitat Berlin. *PIK-Praxis der Informationsverarbeitung und Kommunikation*, 30 (3):145–151, 2007.

[42] Thomas Hildmann. Entwicklung eines sicheren, erweiterten E-Mail-Systems. Diplomarbeit, Fachbereich Informatik, Institut für Kommunikations- und Softwaretechnik, 1998.

[43] Thomas Hildmann. Maßnahmen zum Schutz der Sicherheitspolitik bei der RBAC-Modellierung insbesondere bei der Verwendung von eXtreme Role-Engineering. In Christian Paulsen, editor, *Sicherheit in vernetzten Systemen - 16. DFN Workshopband*. Books on Demand. Norderstedt, 2009.

[44] Thomas Hildmann. Vermeidung von Datenspuren bei smartcard-basierten Authentisierungssystemen. In *Verlässliche IT-Systeme 2001, Sicherheit in komplexen IT-Infrastrukturen*. Vieweg Wiesbaden, 2001. ISBN 3-528-05782-3.

[45] Thomas Hildmann and Jörg Bartholdt. Managing trust between collaborating companies using outsourced role based access control. In *Proceedings of the Fourth ACM RBAC Workshop*, October 1999.

[46] Thomas Hildmann and Thomas Gebhardt. Protecting services with smartcard-based access control: A case study at technical university berlin. In *Proceedings HPOVUA 2001*, June 2001.

[47] Thomas Hildmann and Thomas J. Wilke. Pseudonymous authentication and authorization enhancing ubiquitious identity management. In *Proceedings ISSE 2005*, Budapest, September 26-29 2005.

[48] Thomas Hildmann, Odej Kao, and Christopher Ritter. eXtreme Role Engineering: Ein neuer Ansatz zur Rechtedefinition und -vergabe. In *Proceedings der GI Tagung Sicherheit 2008*, 2008.

[49] Thomas Hildmann, Odej Kao, and Christopher Ritter. Rollenbasierte Identitäts- und Autorisierungsverwaltung an der TU Berlin. In *Proceedings 1. DFN-Forum Kommunikationstechnologien Verteilte Systeme im Wissenschaftsbereich*, 2008.

[50] Trent Jaeger and Jonathon E. Tidswell. Rebuttal to the nist rbac model proposal. In *Fifth ACM Workshop on Role-Based Access Control*. Association for Computing Machinery (ACM), Special Interest Group in Security, Audit, and Control (SIGSAC) and Technical University of Berlin, July 2000.

[51] Heike Jahberg and Barbara Junge. Mehr Datenschutz – nach der Wahl. Tagesspiegel, Februar 2009.

[52] Ron Jeffries. An agile software development resource, November 2008. URL `http://www.xprogramming.com/`.

[53] Anas Abou El Kalam, Rania El Baida, and Philippe Balbiani. Organization based access control. In *Proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*, 2003.

[54] Odej Kao. *Dynamisches Retrieval von multimedialen Daten auf parallelen Architekturen*. Shaker Verlag, 2002.

[55] Neeran Karnik, Girish Cha E, Arun Kumar, and Arun Kumar. Context sensitivity in role-based access control. *In ACM SIGOPS Operating System Review 36*, 36:53–66, 2001.

[56] Asif Mohammad Khan. Modellierung eines Workflow Management Systems innerhalb des rollenbasierten Autorisierungssystems TUBIS. Diplomarbeit, Fakultät IV - Elektrotechnik und Informatik, Technische Universität Berlin, Oktober 2007.

[57] Gregor Kiczales, John Lamping, Anurag Mendhekar, Cristina Videira Lopes, Chris Maeda, Jean-marc Loingtier, and John Irwin. Aspect-oriented programming. In *Lecture Notes in Computer Science 1357*, pages 48–3. Springer Verlag, 1998.

[58] Marit Köhntopp. "Wie war noch gleich Ihr Name?" – Schritte zu einem umfassenden Identitätsmanagement. In Andreas Pfitzmann, editor, *Verlässliche IT-Systeme 2001*, DuD-Fachbeiträge, pages 77–85. Vieweg, September 2001.

[59] J. Kolter, R. Schillinger, and G. Pernul. A privacy-enhanced attribute-based access control system. *Lecture Notes in Computer Science*, 4602:129, 2007.

[60] Martin Kuhlmann, Dalia Shohat, and Gerhard Schimpf. Role mining - revealing business roles for security administration using data mining technology. In *SACMAT '03: Proceedings of the eighth ACM symposium on Access control models and technologies*, pages 179–186, New York, NY, USA, 2003. ACM. ISBN 1-58113-681-1. doi: http://doi.acm.org/10.1145/775412.775435. URL `http://portal.acm.org/ft_gateway.cfm?id=775435&type=pdf&coll=GUIDE&dl=GUIDE&CFID=27664603&CFTOKEN=53632245`.

[61] B. W. Lampson. Dynamic protection structures. *AFIPS Conference Proceedings*, 35: pp. 27–38, 1969.

[62] E. Lüders. Analyse psychischer Belastungen in der Arbeit: Das RHIA-Verfahren. *Handbuch psychologischer Arbeitsanalyseverfahren*, pages 365–395, 1999.

[63] Andreas K. Mattas, Ioannis K. Mavridis, and George I. Pangalos. Towards dynamically administered role-based access control. In *Proceedings of the ninth ACM symposium on Access control*, 2004.

[64] Chrstian Mezler-Andelberg. *Identity Management - eine Einführung.* dpunkt.verlag, Heidelberg, 2008.

[65] Jonathan D. Moffett and Emil C. Lupu. The uses of role hierarchies in access control. In *ACM RBAC Workshop*, Fairfax, VA, USA, October 1999.

[66] Ian Molloy, Hong Chen, Tiacheng Li, Qihua Wang, Ninghui Li, Elisa Bertina, Seraphin Calo, and Jorge Lobo. Mining roles with semantic meanings. In *SACMAT'08*, Colorado, USA, June 2008.

[67] Günter Müller. *Mehrseitige Sicherheit in der Kommunikationstechnik, Bd. 1, Verfahren, Komponenten.* Informationssicherheit. Addision-Wesley, 1997.

[68] Günter Müller. *Mehrseitige Sicherheit in der Kommunikationstechnik, Bd. 2, Erwartung, Akzeptanz, Nutzung.* Informationssicherheit. Addision-Wesley, 1999.

[69] Günter Müller. *Multilateral Security in Communications*, volume 3, Technology, Infrastructure, Economy. Addision-Wesley, 1999.

[70] Gustaf Neumann and Mark Strembeck. Design and implementation of a flexible rbac-service in an object-oriented scripting language. In *In Proc. of the 8th ACM Conference on Computer and Communications Security (CCS)*, 2001.

[71] Gustav Neumann and Mark Strembeck. A scenario-driven role engeneering process for functional rbac roles. In *SACMAT*, 2002.

[72] Oracle. *A Comparison of Oracle Berkeley DB and Relational Database Management Systems*. Oracle, World Headquarters, 500 Oracle Parkway, Redweed Shores, CA 94065, U.S.A., oracle technical white paper edition, November 2006.

[73] Amon Ott. Die Architektur des Linux-Sicherheitssystems Rule Set Based Access Control (RSBAC) - Sicherheits-Architektur. *Linux Magazin*, 01, 2003.

[74] Amon Ott. Wink mit dem Zaunpfahl. *Linux Magazin*, 04, 2003.

[75] Joon S. Park and Ravi S. Sandhu. Rbac on the web by smart certificates. In *RBAC '99: Proceedings of the fourth ACM workshop on Role-based access control*, pages 1–9, New York, NY, USA, 1999. ACM. ISBN 1-58113-180-1. doi: http://doi.acm.org/10.1145/319171.319172. URL `http://portal.acm.org/ft_gateway.cfm?id=319172&type=pdf&coll=GUIDE&dl=GUIDE&CFID=26380430&CFTOKEN=92273267`.

[76] Andreas Pfitzmann. *Multilateral Security in Communications - Technology, Infrastructure, Economy*, volume 3, chapter Technologies for multilateral security, pages 85–91. Addision-Wesley, 1999.

[77] Aneta Poniszewska-Maranda. Role engineering of information system using extended rbac model. In *WETICE'05, IEEE*, 2005.

[78] Daniel Price and Andrew Tucker. Solaris zones: Operating system support for consolidating commercial workloads. In *LISA '04: Proceedings of the 18th USENIX conference on System administration*, pages 241–254, Berkeley, CA, USA, 2004. USENIX Association.

[79] W. Rankl and W. Effing. Handbuch der Chipkarten. *Carl Hanser Verlag, München, Wien*, 4. überarbeitete und aktualisierte Auflage, 2002.

[80] K. Rannenberg, A. Pfitzmann, and G. Mueller. Sicherheit, insbesondere mehrseitige IT-Sicherheit. *INFORMATIONSTECHNIK UND TECHNISCHE INFORMATIK*, 38:7–10, 1996. URL `http://www.wiiw.de/publikationen/Sicherheitinsbesonderemehrsei.pdf`.

[81] Ralf Reißing. Extremes Programmieren. *Informatik Spektrum*, 23(2):118–121, April 2000.

[82] A. Rhodes, W. Caelli, and B. AUSTRALIA. A review paper role based access control. *University of Queensland, Brisbane Australia*, 1999. URL `http://www.isi.qut.edu.au/research/publications/technical/qut-isrc-tr-1999-004.pdf`.

[83] Ravi Sandhu. Future directions in role-based access control models. MMM-ACNS, 2001.

[84] Ravi Sandhu, Venkata Bhamidipati, Edward Coyne, Srinivas Ganta, and Charles Youman. The arbac97 model for role-based administration of roles. *ACM Transactions on Information and System Security*, 2:105–135, 1999.

[85] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *Computer*, Volume 29(2):38–47, February 1996.

[86] Jürgen Schlegelmilch and Ulrike Steffens. Role mining with orca. In *SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 168–176, New York, NY, USA, 2005. ACM. ISBN 1-59593-045-0. doi: http://doi.acm.org/10.1145/1063979.1064008. URL `http://portal.acm.org/ft_gateway.cfm?id=1064008&type=pdf&coll=GUIDE&dl=GUIDE&CFID=27665033&CFTOKEN=95885731`.

[87] Bruce Schneier. *Secret & Lies - IT-Sicherheit in einer vernetzten Welt.* dpunkt.verlag, 2001.

[88] A. Sciberras. Lightweight Directory Access Protocol (LDAP): Schema for User Applications. RFC 4519 (Proposed Standard), June 2006. URL `http://www.ietf.org/rfc/rfc4519.txt`.

[89] SecurityPatterns.org. Security patterns homepage.

[90] Martin Seiler. Identity-managment: Novell vor sun. Computerwoche online, Januar 2006.

[91] Christian Seybold, Silvio Meier, and Martin Glinz. Scenario-driven modeling and validation of requirements models. In *SCESM '06: Proceedings of the 2006 international workshop on Scenarios and state machines: models, algorithms, and tools*, pages 83–89, New York, NY, USA, 2006. ACM. ISBN 1-59593-394-8. doi: http://doi.acm.org/10.1145/1138953.1138969.

[92] Michael E. Shin and Gail-John Ahn. Uml-based representation of role-based access control. In *Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 195 – 200, 2000.

[93] Skip Slone and The Open Group Identity Management Work Area. Identity management. A whitepaper, The Open Group, 44 Montgomery St. 960, San Francisco, CA 94104, March 2004.

[94] Lutz Suhrbier and Thomas Hildmann. Pki based access control with attribute certificates for data held on smartcards. In *Proceedings HPOVUA 2002*, 2002.

[95] Jaideep Vaidya, Vijayalakshni Atluri, and Janice Warner. Roleminer: Mining roles using subset enumeration. In *CCS'06*, Alexandria, Virginia, USA, October - November 2006. ACM.

[96] Jaideep Vaidya, Vijayalakshmi Atluri, and Qi Guo. The role mining problem: Finding a minimal descriptive set of roles. In *SACMAT'07*. ACM, June 2007.

[97] Axel van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *what, why, where and when questions about this area of Requirements EngineeringProceedings RE'01, 5th IEEE International Symposium on*, pages 249–263, August 2001.

[98] W. Volpert. *Wie wir handeln-was wir können*. Artefact-Verlag Weber, 1999.

[99] H. Wegener. Extreme Ansichten. Für und Wider des Extreme Programming. *iX Journal. Heise Verlag*, 12, 1999.

[100] Steffen Weiß and Klaus Meyer-Wegener. Towards solving the data problem in measurement of organizations' security. In Ammar Alkassar and Jörg Siekmann, editors, *Sicherheit 2008 - Sicherheit, Schutz und Zuverlässigkeit*, GI-Edition, pages 461–472. Gesellschaft für Informatik e.V. (GI), April 2008.

[101] Wikipedia. Hamming-Abstand — Wikipedia, Die freie Enzyklopädie, 2008. URL `http://de.wikipedia.org/w/index.php?title=Hamming-Abstand&oldid=47668123`. [Online; Stand 6. Juli 2008].

[102] Wikipedia. Identitätsmanagement — Wikipedia, Die freie Enzyklopädie, 2009. URL `http://de.wikipedia.org/w/index.php?title=Identit%C3%A4tsmanagement&oldid=56693983`. [Online; Stand 5. März 2009].