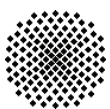


**Entwicklung eines
Rollenmodells zur
nachhaltigen Unterstützung
der Forschung und Lehre
im Bereich Kerntechnik**

Andreas Piater



**Entwicklung eines
Rollenmodells zur
nachhaltigen Unterstützung
der Forschung und Lehre
im Bereich Kerntechnik**

von der Fakultät Energie-, Verfahrens-
und Biotechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

vorgelegt von

Dipl.-Ing. Andreas Piater

geboren in Schwäbisch-Hall.

Hauptberichter:

Prof. Dr.-Ing. F. Schmidt

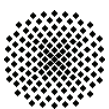
Mitberichter:

Prof. Dr.-Ing. M. Resch

Tag der Einreichung: 12.10.2007

Tag der mündlichen Prüfung: 01.04.2008

ISSN – 0173 – 6892



Kurzfassung

Das Wissen im Bereich Kerntechnik ist auf viele Forschungsorganisationen, Herstellerbetriebe, Energieversorgungsbetriebe, Abfallentsorgungsunternehmen, Genehmigungsbehörden und Hochschulen verteilt. Dort existiert eine Vielzahl an Systemen (Datenbanken, Informationssysteme, Simulationsprogramme, Lehr-/Lernsysteme), die als Teilsysteme in eine gemeinsame Wissensbasis integriert werden könnten. Eine solche Integration ist in der Praxis schwierig, da das Teilwissen über viele Systeme und Köpfe verteilt ist. Sie ist nur dann möglich, wenn Schnittstellen geschaffen werden, die als gemeinsamer Nenner von allen Systemen genutzt und angewandt werden, über welche das Wissen ausgetauscht werden kann. Dadurch könnten alle beteiligten Interessengruppen einen Mehrwert schöpfen.

Eine Integration der verteilten Wissensquellen in eine gemeinsame Wissensbasis ist aber nur dann möglich, wenn gleichzeitig die Sicherheit der Infrastruktur und die Eigentumsrechte an Programmen und Daten beteiligter Interessengruppen gewahrt werden. Um dies sicherzustellen, müssen entsprechende Modelle und Methoden der Zugriffssteuerung entwickelt werden.

Im Rahmen der vorliegenden Arbeit wird dazu ein Benutzermodell mit Rollen und Rechten in Form eines speziellen Role-Based View Control (RBVC)-Rollenmodells für die Kerntechnik entwickelt. Es erweitert das ANSI-genormte Role-Based Access Control (RBAC)-Referenzmodell um verbesserte Integrationsfähigkeiten unterschiedlicher Quellobjekte, erweiterte Rollenhierarchien, verteilte Administration und um die direkte Kopplung der reinen Zugriffssteuerung mit der Steuerung rollenbasierter Sichten. Eine Sicht repräsentiert eine Funktion eines Systems in einer bestimmten Ausprägung. Diese wird unter besonderer Berücksichtigung der Erfordernisse der Kerntechnik an die Aufgabe und den Kenntnisstand eines typischen Benutzers und somit an dessen Rolle angepasst und optimiert. Die Ergebnisse können leicht auf andere Ingenieurdisziplinen übertragen werden.

Im zweiten Teil der Arbeit wird eine Basisarchitektur für web-basierte Systeme auf Grundlage des RBVC-Modells vorgestellt. Es handelt sich um eine Dreischichtarchitektur, deren zentrale Logikschicht durch austauschbare Komponenten erweitert werden kann. Die Architektur erlaubt die Umsetzung kerntechnischer Integrations- und Informationssysteme, in welche unterschiedliche Quellen eingebunden werden

können. Dabei ist kein räumlicher Bezug notwendig, eine Integration kann auch als Web-Service über das Internet erfolgen.

Das RBVC-Rollenmodell und die Basisarchitektur werden in verschiedenen kern-technischen Systemen angewandt und umgesetzt, um deren Tragfähigkeit zu überprüfen. Dabei zeigt sich, dass durch die Kopplung der Zugriffssteuerung mit rollenbasierten und optimierten Sichten viele neue Kombinationsmöglichkeiten geschaffen werden. Simulationsprogramme, deren Integration für die Forschung vorgenommen wurde, können in vereinfachenden Sichten auch für die Lehre verwendet werden. Diese Mehrfachverwendung eröffnet eine neue Dimension, indem der Memorisierungsgrad einer Lehrveranstaltung verbessert und somit deren Nachhaltigkeit erreicht werden kann.

Abstract

In nuclear technology, knowledge is distributed amongst many different organizations – research centers, vendors, utilities and waste management organizations, regulatory bodies, technical safety organizations, and the academia. These organizations comprise a vast number of systems (data bases, information systems, simulation programs and learning environments) which in principle could be integrated as part of a common knowledge base. However, in practice this integration is difficult because the partial knowledge is distributed amongst many systems and persons. Only the creation of new interfaces – as common denominators of all systems – will allow for a knowledge transfer that all stakeholders can benefit from.

The integration of a variety of individual sources of knowledge into one common knowledge base is only possible if the security of infrastructures and the intellectual property (programs and data) are guaranteed and remain in the hands of the stakeholders. Therefore new models and methods for access control have to be developed.

This thesis describes a new role model called Role-Based View Control (RBVC), which extends the standardized Role-Based Access Control (RBAC) ANSI reference model. The new model introduces a better integration of different source objects, extended role hierarchies, distributed administration and the direct coupling of access and view control. One view describes the behavior of a system function in a special characteristic that is optimized for the duties and the level of knowledge of a typical user in a respective role, taking into account the demands of nuclear technology. The resulting model can easily be transferred to other disciplines of engineering.

The second part of the thesis introduces a new base architecture for web-based systems that implements the new RBVC model. It is based on a three-tier architecture with a central logic tier containing replaceable components that are used for extension. Nuclear integration and information systems based on different sources can be built on this new architecture. There are no regional restrictions because integration can also be achieved by web services via the internet.

The RBVC role model and the base architecture have been applied and implemented in several nuclear systems to examine their behavior in practice. It shows that the coupling of access control with role-based and optimized views creates an extra

number of possible combinations. Simulations programs that have been integrated for research can, with simplified views, also be used for teaching. These multiple uses open a new dimension of improving the memorization of courses – and therefore their sustainability.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Gespeichertes und abrufbares Wissen	3
1.2	Lehr-/Lernsysteme und Trainingssysteme	4
1.3	Simulationsprogramme	5
1.4	Rollenmodelle.....	8
1.5	Ziel der Arbeit	10
1.6	Fallbeispiele	14
1.6.1	Fallbeispiel Simulationsprogramm - Wkind	14
1.6.2	Fallbeispiel Informationssystem - Sinter Network.....	15
1.6.3	Fallbeispiel Einsatz von ABR-KFUE als Forschungs- und Trainingssystem	16
2	Grundlagen.....	18
2.1	Stand der Technik	18
2.1.1	Entwicklung der Bewertung der Zugriffssteuerungsmethoden	18
2.1.2	Referenzmodelle für die Zugriffssteuerung	19
2.2	RBAC-Referenzmodell	21
2.2.1	Begriffsdefinitionen.....	21
2.2.2	Basis-RBAC-Referenzmodell	22
2.2.3	Hierarchisches RBAC	25
2.2.4	Aufgabenteilung	27
2.2.5	Paketauswahl bei RBAC	30
2.3	Kritik am bestehenden RBAC-Referenzmodell.....	31
2.4	Erweiterungen durch die Kopplung der Sichtensteuerung mit dem Rollenmodell.....	33
2.5	Vorgehensmodell zur Entwicklung des RBVC.....	34
3	Rollenmodell für den Fachbereich Kerntechnik	35
3.1	Szenarien für den Einsatz eines Rollenmodells	35
3.1.1	Szenario: Simulationsplattform für Forschung und Lehre	35
3.1.2	Szenario: Integrationssystem	35
3.1.3	Szenario: Fernauthentifizierung und -autorisierung.....	36
3.1.4	Szenario: Kerntechnisches Informationssystem.....	36

3.1.5	Szenario: Kombinationssystem – gemeinsame kerntechnische Wissensbasis	36
3.2	Funktionsanforderungen des Rollenmodells	37
3.3	Überführung des RBAC-Modells in ein RBVC-Modell	37
3.3.1	Begriffsdefinitionen.....	37
3.3.2	Ressourcentyp	38
3.3.3	Rollentyp	40
3.4	Rollenbasierte Sichtensteuerung.....	41
3.4.1	Was ist rollenbasierte Sichtensteuerung?	41
3.4.2	Funktionsauswahl durch den Benutzer	43
3.4.3	Fachkompetenzen und zugewiesene Aufgaben.....	45
3.5	Verteilte Administration	47
3.6	Authentifizierung.....	49
3.6.1	Authentifizierung durch Wissen.....	49
3.6.2	Authentifizierung durch Besitz.....	50
3.6.3	Authentifizierung durch biometrisches Merkmal.....	51
3.6.4	Authentifizierung in Bezug auf das RBVC-Modell	52
3.7	RBVC-Gesamtmodell	53
4	Konzept und Implementierung einer Basisarchitektur für Web-Anwendungen unter Berücksichtigung des RBVC-Modells.....	55
4.1	Definition der Funktionsanforderungen	55
4.2	Mehrschichtarchitektur	56
4.2.1	Präsentationsschicht	58
4.2.2	Logikschicht	60
4.2.3	Datenschicht	62
4.3	Programmiersprache und Entwicklungsumgebung	62
4.3.1	Java	63
4.3.2	Eclipse	63
4.4	Framework für die Logikschicht.....	63
4.4.1	Separation of Concerns	64
4.4.2	Servlet.....	64
4.4.3	Control Flow mit Continuations	66
4.4.4	Cocoon Forms.....	66
4.4.5	Sitemaps und Pipelines.....	68

4.4.6	Komponenten in Apache Cocoon.....	70
4.5	Optionale Ergänzung der Logikschicht durch Google Web Toolkit.....	72
4.6	Realisierung der Datenschicht.....	73
4.6.1	Persistenz-Framework	74
4.6.2	Relationale Datenbank.....	74
4.7	Implementierung der Komponenten der Basisarchitektur.....	74
4.7.1	RbacManager.....	75
4.7.2	AuthenticationManager	75
4.7.3	MenueManager	76
4.7.4	ViewControlManager und ViewGenerationManager	76
4.7.5	SourceManager.....	76
4.7.6	RemoteAuthorizationManager und RemoteAccessManager	76
4.7.7	RemoteApplicationManager und RemoteDataObjectManager	77
4.7.8	Test der Komponenten.....	78
5	Anwendungen des Rollenmodells	80
5.1	Fallbeispiel Wkind und Integration in eine Lehr-/Lernumgebung.....	80
5.2	Fallbeispiel SINTER Network und Überführung in SINTER XT.....	84
5.3	Fallbeispiel ABR-KFUE und Überführung in ABR-Research	86
5.4	NEPTUNO-CS.....	87
5.5	GRS-FBW	89
6	Zusammenfassung	92
7	Ausblick	95
	Literaturverzeichnis	96

Abbildungsverzeichnis

Abb. 1 :	Kerntechnischer Wissenskreis – Wissensproduktion, Wissensverbreitung und Wissensverwertung [5].....	2
Abb. 2 :	Modellbildung und Simulation	7
Abb. 3 :	Einfaches Rollenmodell.....	10
Abb. 4 :	Zuordnung von Berechtigungen zu Benutzern nach rollen- (RBAC) und gruppenbasierten Modellen [27]	20
Abb. 5 :	Core RBAC Model nach ANSI INCITS 359-2004 [28].....	22
Abb. 6 :	Rollenhierarchien [28]	25
Abb. 7 :	Beispiel eines Hasse-Diagramms mit Rollen, Vererbung und Benutzern	26
Abb. 8 :	Statische Aufgabenteilung – Static Separation of Duty [28].....	28
Abb. 9 :	Beispiel einer statischen Aufgabenteilung.....	28
Abb. 10 :	Dynamische Aufgabenteilung – Dynamic Separation of Duty [28]	30
Abb. 11 :	Methode um funktionale RBAC-Pakete zusammenzustellen [28]	31
Abb. 12 :	Erweiterung des RBAC-Modells um Ressourcentypen	38
Abb. 13 :	Erweiterung des RBAC-Modells um Rollentypen	40
Abb. 14 :	Erweiterung der Hierarchie-Arten mit Hilfe der Rollentypen.....	40
Abb. 15 :	RBVC-Modell als Erweiterung des RBAC-Modells.....	42
Abb. 16 :	Exemplarischer Aufbau einer Oberfläche für rollenoptimierte Sichten .	43
Abb. 17 :	Beispiel eines Menüs mit kollidierenden Rollen für einzelne Sichten...	44
Abb. 18 :	Einheiten zur Administration von RBVC.....	48
Abb. 19 :	RBAC-Referenzmodell (schwarz) und Erweiterungen (rot) zur Überführung in das RBVC-Gesamtmodell.....	53
Abb. 20 :	Dreischichtmodell der Basisarchitektur mit Unterstützung von RBVC (konventionelle Darstellung)	57
Abb. 21 :	Dreischichtmodell der Basisarchitektur mit Unterstützung von RBVC (direkte Abbildung des Rollenmodells)	57
Abb. 22 :	Vergleich der Klient-Server-Interaktion bei klassischer Web-Anwendung und bei AJAX-Web-Anwendung [54]	58
Abb. 23 :	Vergleich synchroner und asynchroner Kommunikation [54]	60
Abb. 24 :	Separation of Concerns [76].....	64
Abb. 25 :	Apache Web-Server als Reverse-Proxy-Server mit Lastverteilung	65

Abb. 26 :	Ablauf einer Formuldarstellung mit Cocoon Forms [86]	67
Abb. 27 :	Pipeline verarbeitet einen Control Flow-Aufruf durch eine URL	69
Abb. 28 :	Feste Kopplung ohne Inversion of Control (vgl. [91])	71
Abb. 29 :	Lose Kopplung durch Inversion of Control (vgl. [91])	71
Abb. 30 :	Aufbau der Google Web Toolkit Remote-Procedure-Calls [96]	72
Abb. 31 :	Zugriff eines räumlich getrennten Systems über das Internet	77
Abb. 32 :	Zugriff auf räumlich getrennte Anwendungen und Datenobjekte über das Internet	78
Abb. 33 :	Sicht für Studierende: Parameterformular des Simulations- programms Wkind	83
Abb. 34 :	Sicht für Studierende: Visualisierung eines Teilergebnisses des Simulationsprogramms Wkind	84
Abb. 35 :	Übernahme von bisher nicht gepflegten Kursdaten durch eine Organisation	88
Abb. 36 :	Auswahl der Kriterien zur Suche nach veröffentlichten Fortschrittsberichten in GRS-FBW	90

Tabellenverzeichnis

Tab. 1 :	Hierarchie-Arten	41
Tab. 2 :	Vor- und Nachteile der Authentifizierung durch Wissen	50
Tab. 3 :	Vor- und Nachteile der Authentifizierung durch Besitz	51
Tab. 4 :	Vor- und Nachteile der Authentifizierung durch biometrische Merkmale.....	52
Tab. 5 :	Zuweisung der Rollen zu optimierten Sichten	81

Abkürzungsverzeichnis

ABR-KFUE *Ausbreitungsrechnung in der Kernreaktor-Fernüberwachung*

AJAX *Asynchronous JavaScript and XML*

ANSI *American National Standards Institute*

API *Application Programming Interface*

BSI *Bundesamt für Sicherheit in der Informationstechnik*

CASTOR *Cask for Storage and Transport of Radioactive Material*

CC *Common Criteria for Information Technology Security Evaluation*

CForms *Cocoon Forms*

CSS *Cascading Style Sheets*

CTCPEC *Canadian Trusted Computer Product Evaluation Criteria*

DAC *Discretionary Access Control*

DoD *Department of Defence*

DSD *Dynamic Separation of Duty*

EA *Endlicher Automat*

ENEN *European Nuclear Education Network*

ERA *European Research Area*

EU *Europäische Union*

FP-5 *5th Framework Programme*

FP-6 *6th Framework Programme*

FP-7 *7th Framework Programme*

FSM *Finite State Machine*

GRS *Gesellschaft für Anlagen- und Reaktorsicherheit mbH*

GRS-FBW *GRS-Forschungsberichtswesen*

GWT *Google Web Toolkit*

HQL *Hibernate Query Language*

HTML *Hypertext Markup Language*

HTR *Hochtemperaturreaktor*

http *Hypertext Transfer Protocol*

https *Hypertext Transfer Protocol Secure*

I18n *Internationalisierung*

IDE *Integrated Development Environment*

IoC *Inversion of Control*

IP *Integrated Projects*
ISO *International Organization for Standardization*
ITSEC *Information Technology Security Evaluation Criteria*
ITSK *Informationstechnik-Sicherheitskriterien*
J2EE *Java 2 Platform Enterprise Edition*
JVM *Java-Virtual-Machine*
MAC *Mandatory Access Control*
MVC *Model View Controller*
NEPTUNO *Nuclear European Platform for Training and UNiversity Organisations*
NEPTUNO CS *Nuclear European Platform for Training and UNiversity Organisations
Communication System*
NoE *Networks of Excellence*
OBS *Objects*
OPS *Operations*
PA *Permission Assignment*
PIN *Personal Identification Number*
POJO *Plain Old Java Object*
PRMS *Permissions*
R&D *Research & Development*
RAPHEL *ReActor for Process heat, Hydrogen And ELectricity generation*
RBAC *Role-Based Access Control*
RBVC *Role-Based View Control*
RFID *Radio Frequency Identification*
RH *Role Hierarchy*
RPC *Remote-Procedure-Call*
SAX *Simple API for XML*
SCRAM *Safety Control Rod Axe Man*
SINTER *Sustainable & Innovative Nuclear Technology Evolution – R&D*
SINTER XT *Sustainable & Innovative Nuclear Technology Evolution – R&D
eXtended Technology*
SoC *Separation of Concerns*
SoD *Separation of Duty*
SQL *Structured Query Language*
SSD *Static Separation of Duty*

SSL *Secure Sockets Layer*
TCSEC *Trusted Computer System Evaluation Criteria*
UA *User Assignment*
UH *Unit Hierarchy*
URA *Unit-Role Assignment*
URL *Unified Resource Locator*
VA *View Assignment*
WBT *Web-based Training*
XML *Extensible Markup Language*
XSL *Extensible Stylesheet Language*
XSLT *Extensible Stylesheet Language Transformations*

1 Einleitung

Der Europäische Rat formulierte auf seiner Tagung in Lissabon im Jahr 2000 das strategische Ziel, die Europäische Union zur wettbewerbsfähigsten und dynamischsten wissensbasierten Wirtschaft der Welt zu machen:

„to become the most competitive knowledge-based economy with more and better employment and social cohesion by 2010“. [1]

Um dieses Ziel zu erreichen, müssen vor allem in den Bereichen Forschung, Entwicklung und Lehre neue Methoden und Modelle entwickelt werden. Diese sollen helfen, bestehendes Wissen zu bewahren und unterstützend dazu beitragen, das explizite Wissen als zukünftige Basis neuen Wissens effizient nutzen zu können. Im Vertrag zur Gründung der Europäischen Gemeinschaft von Amsterdam steht hierzu in Artikel 163:

„Die Gemeinschaft hat zum Ziel, die wissenschaftlichen und technologischen Grundlagen der Industrie der Gemeinschaft zu stärken und die Entwicklung ihrer internationalen Wettbewerbsfähigkeit zu fördern sowie alle Forschungsmaßnahmen zu unterstützen, die aufgrund anderer Kapitel dieses Vertrags für erforderlich gehalten werden“. [2]

Die Europäische Kommission hat von 1998 bis 2002 mit dem 5. Rahmenprogramm¹ (FP-5) [3] den Wissensaufbau im Bereich Kerntechnik vorangetrieben. Als Resultat entstand fragmentiertes Wissen, das auf die beteiligten Interessengruppen, bestehend aus Forschungsorganisationen, Herstellerbetrieben, Energieversorgungsbetrieben, Abfallentsorgungsunternehmen, Genehmigungsbehörden und Hochschulen verteilt war. Da das Wissen individuell, zielorientiert und daher mit unterschiedlicher Systematik erworben wurde, bildete sich keine gemeinsame Wissensbasis und ein Zugriff auf diese Informationen war nur teilweise und nicht konsistent möglich.

Erklärtes Ziel des 6. Rahmenprogramms² (FP-6) [4] [5] seit 2003 ist es daher, innerhalb des Europäischen Forschungsraums³ (ERA) [6] die Voraussetzungen für eine gemeinsame und dauerhafte Wissensbasis zu schaffen. Das Startkapital wird durch FP-6 bereitgestellt und soll als Katalysator dienen, den Integrationsprozess in den

¹ 5th Framework Programme

² 6th Framework Programme

³ European Research Area

drei Bereichen *Wissensproduktion*, *Wissensverbreitung* und *Wissensverwertung* anzustoßen.

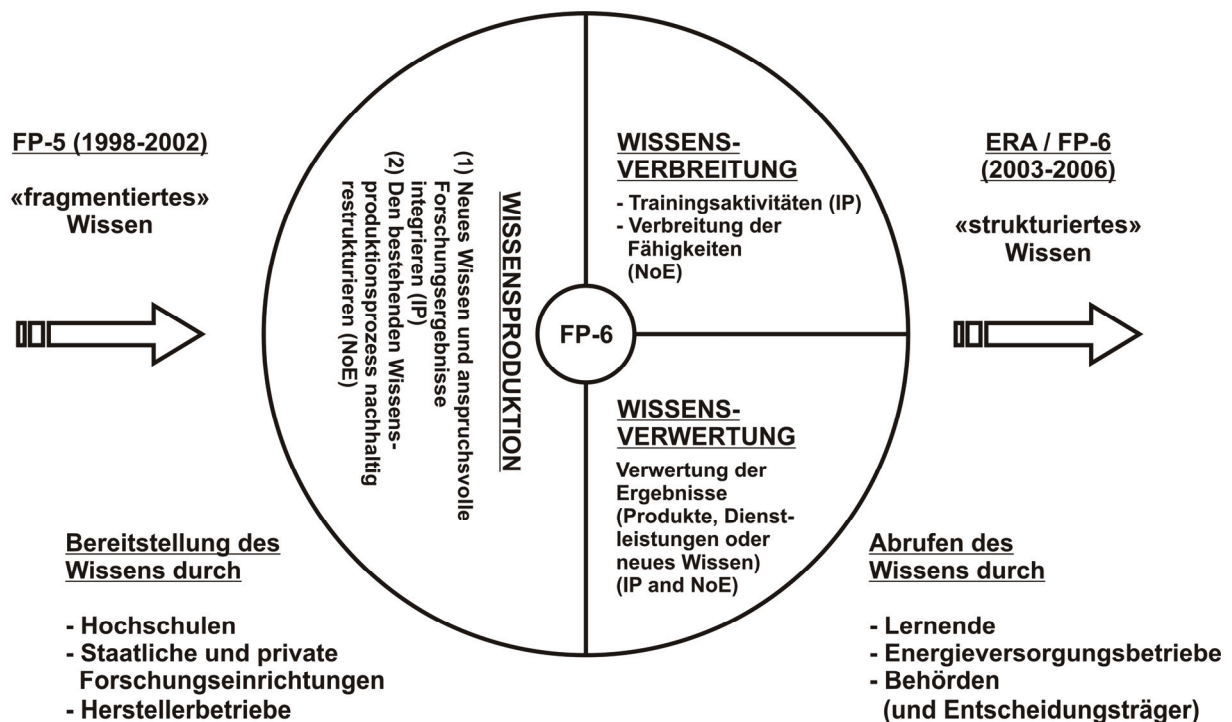


Abb. 1: Kerntechnischer Wissenskreis – Wissensproduktion⁴, Wissensverbreitung⁵ und Wissensverwertung⁶ [5]

Der kerntechnische Wissenskreis (Abb. 1) verdeutlicht diese zentrale Funktion des FP-6. Das verteilte Wissen aus FP-5 soll durch die drei oben genannten Wissensbereiche in eine strukturierte Form übergeführt werden. Dieser Vorgang wird durch die in FP-6 neu eingeführten *Integrierten Projekte*⁷ (IP) [7] und *Exzellenznetzwerke*⁸ (NoE) [8] unterstützt. Um das Ziel des FP-6 zu erreichen, müssen die Wissenslieferanten und Wissensabnehmer eine gemeinsame Strategie verfolgen und gemeinsame Werkzeuge zum Wissensaustausch einsetzen.

Das kommende 7. Rahmenprogramm⁹ (FP-7) setzt die Ziele aus FP-6 nahtlos fort. Im Beschluss des Europäischen Parlaments und des Rates aus dem Jahr 2006 zur Einrichtung des FP-7 werden unter dem Programm „unternehmerische Initiative und Innovation“ als Innovationstätigkeiten folgende Ziele und Aktionsbereiche genannt:

⁴ Production

⁵ Dissemination

⁶ Exploitation

⁷ Integrated Projects

⁸ Networks of Excellence

⁹ 7th Framework Programme

„Unterstützung von Diensten für den transnationalen Wissens- und Technologietransfer und für den Schutz und die Verwaltung des geistigen und gewerblichen Eigentums;“ [9]

und weiter

„Förderung des Wissenstransfers durch Archivierung und Transfer von Daten.“ [9]

Die Ziele aus FP-6, eine gemeinsame, dauerhafte kerntechnische Wissensbasis aufzubauen, und aus FP-7, den Wissenstransfer zu fördern, können dadurch erreicht werden, dass spezielle internetbasierte Softwaresysteme die Kooperation und die Kohärenz der europäischen Interessengruppen unterstützen. Eine dafür notwendige Harmonisierung des Wissens ist aber nur möglich, wenn gleichzeitig die Sicherheit bestehender Infrastrukturen der Kompetenzträger berücksichtigt wird.

Das zu harmonisierende Wissen, welches die zukünftige Wissensbasis aufnehmen muss, lässt sich idealtypisch in drei große Bereiche unterteilen:

1. Gespeichertes und abrufbares Wissen
2. Lehr-/Lernsysteme und Trainingssysteme
3. Simulationsprogramme

In den folgenden Unterkapiteln werden die drei Bereiche näher beschrieben.

1.1 Gespeichertes und abrufbares Wissen

Das Wissen im Bereich der Kerntechnik ist über die Forschungsorganisationen, Herstellerbetriebe, Energieversorgungsbetriebe, Abfallentsorgungsunternehmen, Genehmigungsbehörden und Hochschulen verteilt. Klassisch ist das Wissen statisch strukturiert und lässt sich in Bereiche mit entsprechender Zugänglichkeit unterteilen:

- Private Sammlungen (nicht öffentlich zugänglich)
- Bibliotheken (meist öffentlich zugänglich)
- Archive (meist nicht öffentlich zugänglich)

- Datenbanken, Wissensbanken und Informationssysteme im Intranet (nicht öffentlich zugänglich)
- Datenbanken, Wissensbanken und Informationssysteme im Internet (öffentlich zugänglich, teilweise zugriffsgeschützt)

Die Bereiche enthalten Wissen über Daten (Materialdaten, Versuchsergebnisse, etc.) und existierendes explizites Wissen (Forschungsberichte, Knowledge-Bases, etc.), das im Laufe der Jahre aus Forschung und Entwicklung gewonnen wurde.

Die einzelnen Bereiche sind nicht oder nur teilweise miteinander verknüpft. Sofern das Wissen elektronisch vorliegt, können es die Benutzer meist nur räumlich begrenzt in Intranets abrufen. Ein Austausch über die Interessengruppen hinweg ist oft erschwert, da die bestehenden Softwaresysteme dies nur selten zulassen. Wenn sie es zulassen, dann nur in einem begrenztem Umfang. Ursächlich hierfür sind Sicherheitsaspekte, Wert der Daten für eine Organisation, geistige Eigentumsrechte und häufig auch eine Inkompatibilität der eingesetzten speziellen Softwaresysteme. Diese bieten meist keine standardisierten Schnittstellen nach außen an, auf die über ein gemeinsames Benutzermodell zugegriffen werden könnte.

1.2 Lehr-/Lernsysteme und Trainingssysteme

Im Rahmen der Ausbildung an Hochschulen werden heute verschiedene Lehr-/Lernsysteme unterstützend zu Vorlesungen eingesetzt. Diese dienen der Wissensaneignung und der Übung. Sie sollen den Memorierungsgrad erhöhen und ein tieferes und besseres Verständnis des zu vermittelnden Stoffes erreichen. Sie lassen sich dadurch von den heute häufig anzutreffenden reinen Sammlungen von Vorlesungsfolien und -skripten abgrenzen, indem sie nicht nur Inhalte zum Herunterladen anbieten, sondern mit ihnen

- die Zugriffe über ein Benutzermodell gesteuert werden können.
- Vorlesungen und Übungen eingestellt werden können.
- der Vorlesungs- und Übungsstoff verwaltet werden kann.

- ergänzend Kommunikationsmethoden und Werkzeuge angeboten werden, die das Lernen unterstützen.
- die Möglichkeit besteht, das Wissen für Schulungszwecke aufzubereiten.

Neben dem hochschulinternen Einsatz von Lehr-/Lernsystemen bietet sich im Bereich Kerntechnik die Anwendung solcher Systeme immer dann an, wenn Ausbildungsvorhaben im europäischen oder internationalen Rahmen stattfinden. Beispielsweise bietet die European Nuclear Education Network Association (ENEN) einen europäischen Studiengang an, der als Abschluss einen „European Master of Science in Nuclear Engineering“ [10] vorsieht.

Der Hauptfokus bei Trainingssystemen liegt im Gegensatz zu Lehr-/Lernsystemen nicht in der grundlegenden Wissensaneignung, sondern in der Übung operativer Fähigkeiten bereits erlernten Wissens. Dabei integrieren diese Systeme Wissensbanken und Simulationsprogramme mit Trainingseinheiten. Sie kommen bei Herstellerbetrieben, Energieversorgungsbetrieben und Abfallentsorgungsunternehmen zum Einsatz, um spezifisches Wissen über Prozesse und Anlagenverhalten zu vermitteln und Mitarbeiter für die operative Arbeit zu schulen. Aber auch bei Genehmigungsbehörden finden unter Verwendung von Trainingssystemen regelmäßige Übungen statt, die ein routiniertes Verhalten der Mitarbeiter im Ereignisfall gewährleisten sollen.

1.3 Simulationsprogramme

Simulationsprogramme konzentrieren Wissen aus unterschiedlichen Quellen. Sie stellen ein wichtiges Hilfsmittel für die Ingenieurdisziplinen dar und werden sehr häufig in den Gebieten Wissensvermittlung, Forschung und Entwicklung eingesetzt. Im Bereich der Wissensvermittlung dienen sie dazu, dynamische Vorgänge, Wechselwirkungen und Auswirkungen von Änderungen zu untersuchen. Im Bereich der Forschung und Entwicklung tragen sie zur Lösung komplexer Fragestellungen anhand numerischer Modelle bei. Dort werden Simulationsprogramme hauptsächlich in Bereichen eingesetzt, in denen eine andere Lösung der zugrunde liegenden mathematischen Modelle zu komplex oder zu aufwendig ist und Versuche an realen Systemen oder Anlagen zu kostenintensiv oder zu gefährlich sind. Auch analytisch unlösbare Problemstellungen lassen sich mit Hilfe von Methoden wie beispielsweise dem Differenzenverfahren, dem Finite-Elemente-Verfahren oder der Monte-Carlo-Methode [11]

numerisch lösen. Zusammenfassend besteht der Zweck einer Simulation entweder darin, das Verhalten eines bekannten Szenarios besser zu verstehen, nachzuvollziehen und optimieren zu können, oder aber das Verhalten von Prozessen und Anlagen vorherzusagen.

Die folgende Aufzählung zeigt beispielhaft Bereiche, in denen Simulationsprogramme sowohl in der Forschung und Entwicklung, aber auch in der Lehre im Bereich Kerntechnik eingesetzt werden:

- Korrelation verschiedener Bereiche: Gesamtschau statt Einzeleffekt in den Bereichen Forschung und Entwicklung neuer Reaktoren
- Untersuchung von alternativen Lösungen: Variantenkonstruktion bei der Forschung und Entwicklung neuer Reaktorgenerationen, beispielsweise EPR¹⁰, VHTR¹¹
- Optimierung des Betriebs unter aktuellen Randbedingungen: kontinuierliche Verbesserung bestehender Reaktoren
- Untersuchungen in Grenzbereichen: Störfallsimulation in der Forschung und Entwicklung auf dem Gebiet Reaktorsicherheit, Wissensvermittlung im Rahmen von Übungen bei der Hochschulausbildung

In Abb. 2 sind schematisch die Schritte zur Bildung eines Modells für ein spezielles Simulationsprogramm dargestellt. Dies zeigt exemplarisch, aus welchen unterschiedlichen Quellen hier Wissen zusammengebracht wird. Dazu wird ausgehend von der konkreten Problemstellung ein physikalisches Modell entwickelt. Dieses stellt ein vereinfachtes Abbild einer partiellen Wirklichkeit dar und beschreibt sie für den zu untersuchenden Zweck hinreichend genau. Das physikalische Modell wird im nächsten Schritt in ein mathematisches Modell umgewandelt und in einem weiteren Schritt analysiert. Dabei wird es auf Handhabbarkeit und Lösbarkeit untersucht und in ein numerisches Modell übergeführt. Im nächsten Schritt wird aus diesem numerischen Modell das Simulationsprogramm erstellt, das aus einem oder mehreren Modulen besteht und Daten mit dem umgesetzten Modell und den ausführenden Methoden verknüpft. Diese Module werden zur Berechnung spezieller Szenarien herangezogen

¹⁰ Europäischer Druckwasserreaktor - European Pressurized Water Reactor

¹¹ Very High Temperature Reactor

und erzeugen Ergebnisse, die in einem weiteren Schritt analysiert und danach visualisiert und beurteilt werden.

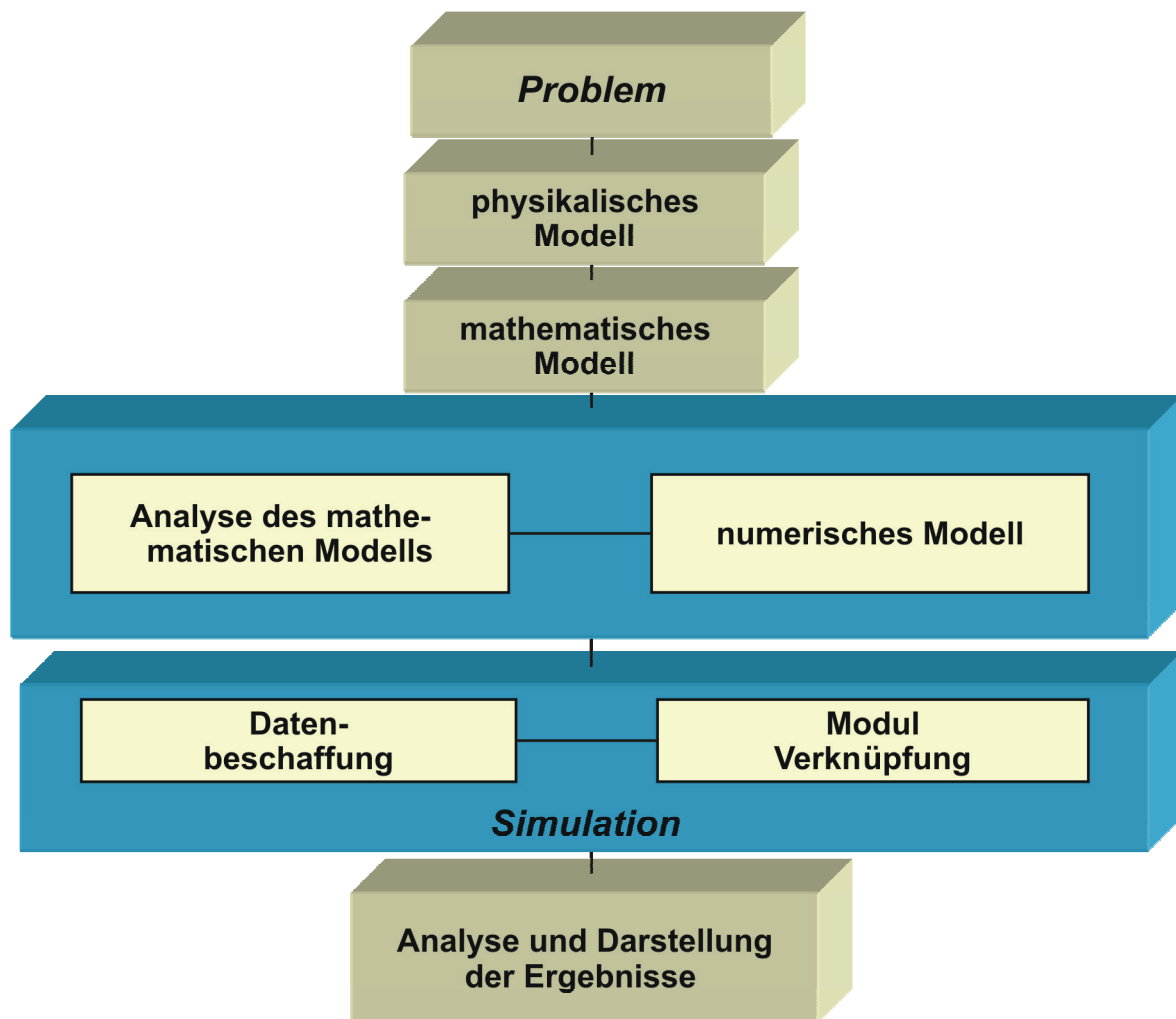


Abb. 2: Modellbildung und Simulation

Neben Neuentwicklungen kann im Fachbereich Kerntechnik auf ein großes Repertoire bestehender Simulationsprogramme zurückgegriffen werden. Sie werden in zahlreichen Disziplinen eingesetzt und dienen zur Auslegung von Reaktoren und dazugehöriger Anlagen.

Die überwiegende Anzahl dieser Programme wurde unter Verwendung der Programmiersprache Fortran entwickelt. Im Rahmen der Programmierung finden hauptsächlich Optimierungen aufgrund begrenzter Hardware-Ressourcen, wie beispielsweise Speicherbedarf und notwendige Prozessorleistung, statt.

Meist kommen bei der Entwicklung der Simulationsprogramme die softwaretechnischen Belange in Bezug auf die zugrundeliegende Programmarchitektur zu kurz. Als Resultat entstehen Programme,

- die ausschließlich auf den jeweiligen Anwendungsfall und auf dessen physikalische Annahmen optimiert werden.
- die aufgrund ihrer Struktur nur eingeschränkt wartbar sind.
- deren Ein- und Ausgabeschnittstellen aus kryptischen, für den ungeübten Anwender schwer verständlichen Text-Dateien¹² bestehen.
- die eine hohe Einarbeitungszeit auch für Experten aufweisen.
- die keine konsistente Fehlerbehandlung durchführen.
- die alleine aus Kostengründen momentan nicht durch Neuentwicklungen ersetzt werden können.
- deren Validierung nach einer Neuprogrammierung zeitlich und monetär zu kostspielig wäre.
- bei denen das Wissen über die Programme selbst oft schlecht erfasst ist.

1.4 Rollenmodelle

Das Internet ist heute mehr denn je Dreh- und Angelpunkt für den globalen Wissensaustausch. Das Internet kennt keine Schranken. Deswegen müssen bestimmte Informationen vor unberechtigtem Zugriff geschützt werden. Dies gilt besonders für personenbezogene, kontextabhängige oder proprietäre Informationen in sicherheitskritischen Bereichen, wie sie in der Kerntechnik sehr häufig anzutreffen sind. Eine Sicherung kann nur mit einer entsprechenden Zugriffssteuerung¹³ erreicht werden. Beispielsweise müssen im Rahmen des Notfallschutzes der Kernreaktor-Fernüberwachung den Genehmigungsbehörden im Ereignisfall Berechtigungen gegeben werden, die den externen, vollständigen und konsistenten Zugriff auf die sicherheitsrelevanten Informationen des betroffenen Energieversorgungsunternehmens erlauben.

Die Informationen liegen heute in elektronischer Form vor und dürfen nur privilegierten Benutzergruppen zugänglich gemacht werden. Sie sind aber, wie auch das vorherige Beispiel aus dem Bereich Notfallschutz zeigt, auf die einzelnen Interessengruppen verteilt. Wenn auf die dahinterliegenden Datenobjekte ein Zugriff über das Internet gewährt werden soll, dann muss sichergestellt sein, dass durch die Freigabe

¹² Simulationsprogramme, deren Eingabeschnittstellen früher auf Lochkarten basierten, wurden später meist auf Text-Dateien umgestellt.

¹³ Der englische Begriff *Access Control* wird in der deutschen Fachliteratur unter den beiden Begriffen Zugriffssteuerung und Zugriffskontrolle geführt. Im Rahmen dieser Arbeit wird der Begriff Zugriffssteuerung verwendet, da dieser umfassender ist und die Zugriffskontrolle beinhaltet.

die jeweilige Netzwerksicherheit nicht gefährdet wird. Um dieses Ziel zu erreichen, ist es notwendig, dass nur auf diejenigen Daten von berechtigten Personen zugegriffen werden kann, die dafür autorisiert sind und für die spezifische Freigaben bestehen.

Im Bereich des elektronischen Lernens werden für Studierende in integraler Weise Simulationsprogramme, Informationssysteme und Lerneinheiten im Rahmen von Lehr-/Lernsystemen zur Verfügung gestellt. Die Integration stellt sicher, dass die Informationen stets aktuell sind und nicht redundant gehalten werden. Somit ist ein konsistenter Zugriff möglich, der in anwendungsfalloptimierten Sichten¹⁴ dargestellt werden kann. Auf der anderen Seite muss aber sichergestellt werden, dass sowohl Studierende, als auch Lehrer und Tutoren nur auf diejenigen Daten zugreifen können, die für sie vorgesehen sind. Gleiches gilt auch für alle anderen Systeme, die mehrere Quellen integrieren.

Die Sicherheitsanforderungen zeigen, dass besonders bei der Integration verteilter Systeme ein wirksamer, effektiver Zugriffsschutz notwendig ist. Es müssen Methoden bereitgestellt werden, die den Zugriff in solchen Systemen auf spezifische Benutzer und Benutzergruppen beschränken können.

Generell dienen Rollenmodelle dazu, die direkte Abbildung von Benutzern und Benutzergruppen auf Zugriffsberechtigungen bei komplexen Systemen zu entkoppeln. Rollen verstehen sich in diesem Zusammenhang als eine Form abstrakter Benutzergruppen mit gemeinsamen Aufgaben. Unter dem Begriff Rollenmodell wird im Rahmen dieser Arbeit ein Modell verstanden, das aus einem Benutzermodell für ein Softwaresystem mit Rollen und Rechten besteht. Ein solches spezielles Modell für den Fachbereich Kerntechnik zu entwickeln und einzusetzen ist sinnvoll und notwendig, weil hier nicht nur für den Betrieb von Kernkraftwerken, sondern auch für die Forschung, Entwicklung und Lehre sehr hohe Sicherheitsanforderungen gelten müssen, um Missbrauch zu verhindern.

Ein Softwaresystem rein auf Benutzerebene ohne Entkopplung über abstrakte Rollen zu schützen, ist nur mit entsprechend hohem Administrationsaufwand möglich. Allerdings bieten einfache Rollenmodelle, die beispielsweise aus der dreistufigen Struktur Welt, Gruppe und Untergruppe bestehen (Abb. 3) und denen die Benutzer zugewie-

¹⁴ Begriff Sicht im Kontext dieser Arbeit wird in Kapitel 3.3.1 definiert.

sen werden, nur eingeschränkt die Flexibilität, die für mehrere Quellsysteme in einem integrierenden Gesamtsystem benötigt werden.

Welt
Gruppe
Untergruppe

Abb. 3: Einfaches Rollenmodell

Je integraler ein System wird und je mehr Benutzer auf ein System zugreifen, desto größer wird der Aufwand Rollen einzuführen und anzupassen, wenn sich beispielsweise Aufgabengebiete und Berechtigungen einzelner Nutzer ändern. Dadurch entsteht eine große Lücke zwischen dem Aufwand und dem Schutzzweck. Besser sind Rollenmodelle, welche direkt die Aufgaben statt einer Gruppenzugehörigkeit abbilden. Das daraus resultierende Nutzungsverhalten eines Systems kann im Regelfall ohne größeren Aufwand auf Rollen abgebildet werden.

Der Einsatz eines speziellen Rollenmodells für integrierende Systeme im Fachbereich Kerntechnik dient daher dazu, die oben beschriebene Lücke zu schließen und den sehr hohen Anforderungen an die Sicherheit gerecht zu werden. Der Administrationsaufwand verringert sich trotz ausreichend großer Flexibilität. Gleichzeitig erhöht sich die Feingranularität der Berechtigungsmöglichkeiten und somit die Sicherheit. Dadurch werden wichtige Voraussetzungen dafür geschaffen, das Wissen in der Kerntechnik in integraler Form bereitzustellen und für eine zeitgemäße Nutzung aufzubereiten.

1.5 Ziel der Arbeit

Das Ziel dieser Arbeit ist die Entwicklung der Grundlagen für eine Integration der verschiedenen

- Systeme für gespeichertes und abrufbares Wissen aus Kapitel 1.1,
- Lehr-/Lernsysteme und Trainingssysteme aus Kapitel 1.2 und
- Simulationsprogramme aus Kapitel 1.3

dahingehend, dass diese Bestandteil einer gemeinsamen Wissensbasis werden. Dazu wird ein Rollenmodell entwickelt, das auf einer direkten Kopplung eines Zugriffssteuerungsmodells mit einem Sichtenmodell aufbaut, und das eine Harmonisierung des Wissens ermöglicht. Auf dem resultierenden Modell bauen Integrations-

systeme auf, die anwendungsfalloptimierte Sichten sowohl für die Wissenslieferanten (Hochschulen, staatliche und private Forschungseinrichtungen, Herstellerbetriebe), als auch für die Wissenskonsumenten (Lernende, Energieversorgungsbetriebe, Behörden und Entscheidungsträger) zur Verfügung stellen.

Im Gegensatz dazu verteilen bereits existierende Systeme das vorhandene Wissen aus den drei angesprochenen Bereichen über alle Interessengruppen. Sie bestehen meist aus autarken Systemen, weshalb es häufig zu Redundanzen kommt. Selbst innerhalb einer Organisation gibt es voneinander unabhängige Insellösungen, deren Inhalte in redundanter Weise deckungsgleich sind. Da keine infrastruktur- und organisationsübergreifende Zugriffssteuerung existiert, ist ein konsistenter Zugriff auf das fragmentierte Gesamtwissen nicht möglich.

Aus den Einschränkungen der existierenden Systeme lassen sich die Anforderungen an das im Rahmen dieser Arbeit zu entwickelnde Rollenmodell ableiten. Es wird davon ausgegangen, dass der Zugriff auf die Systeme weitgehend über das Internet erfolgt.

Anforderung: Kopplung der Rollen- und Sichtensteuerung

Rollenbasierte Sichten ergeben sich aus einer Verknüpfung zwischen dem zugrundeliegenden Zugriffssteuerungsmodell eines Softwaresystems, das auf einem Rollenmodell basiert, und den Sichten auf das System. Sämtliche Sicherheitsaspekte des Zugriffsteuerungsmodells sind somit auch auf die Sichtendarstellung und -steuerung übertragbar.

Welchen Vorteil bietet die direkte Kopplung der Zugriffssteuerung mit der Sichtendarstellung und -steuerung? Der Ansatz hierfür entsteht aus der Annahme, dass Rollen den Aufgaben entsprechen, die ein typischer Benutzer in einem System einnimmt. Diese Aufgaben setzen einen entsprechenden Kenntnisstand eines typischen Benutzers voraus. Bekommt ein Benutzer eine Rolle zugewiesen, dann wäre eine Aufgaben und Kenntnisstand entsprechende Funktion des Softwaresystems erstrebenswert. Die Funktion muss in diesem Fall in einer speziell optimierten Ausprägung angeboten werden. Diese Funktionsausprägung entspricht einer rollenbasierten Sicht.

Beispiel: Im Rahmen einer gemeinsamen kerntechnischen Wissensbasis können angebundene Lehr-/Lernsysteme direkt auf Systeme für gespeichertes und abrufba-

res Wissen zugreifen, indem für die Lehre eine spezielle Rolle definiert wird, die mit einer aufgabenentsprechenden, optimierten Sicht gekoppelt wird.

Anforderung: Sicherheit

Da für den Aufbau von Integrationssystemen die Sicherheit der Infrastrukturen beteiligter Interessengruppen gewahrt werden muss, basiert das neue Rollenmodell auf bestehenden Referenzmodellen zur Zugriffssteuerung.

Die erhöhten Anforderungen an die Sicherheit sind für den Betrieb kerntechnischer Anlagen obligatorisch und lassen sich auf eine gemeinsame Wissensbasis übertragen, da auch hier Informationen beispielsweise über sicherheitskritische Anlagenteile zu finden sind, die nicht jedem zugänglich gemacht werden dürfen. Der Zugriffsschutz wird zudem durch die Steuerung der Sichten ergänzt, wodurch je nach Anwendungsfall nur eingeschränkte Sichten auf die Informationen gewährt werden. Diese zeigen ausschließlich die Informationen an, welche für die Aufgaben eines typischen Benutzers bezüglich der ihm zugewiesenen Rolle benötigt werden.

Beispiel: Ein Simulationsprogramm, welches das Verhalten eines Reaktorkerns abbildet, wird von einem Herstellerbetrieb in eine gemeinsame Wissensbasis integriert. In Kombination mit entsprechenden Sichten kann das Programm beispielsweise für Untersuchungen der Mitarbeiter des Herstellerbetriebs, für wissenschaftliche Studien von Experten an Hochschulen und für die Ausbildung von Studierenden in der Lehre eingesetzt werden, um dort dynamische Vorgänge untersuchen zu können. Erst die Kombination der Zugriffs- und der Sichtensteuerung in einem gekoppelten Rollenmodell ermöglicht es, je nach Anwendung entscheiden zu können, welcher Benutzer welche Rolle und damit welche mehr oder weniger einschränkende Sicht erhält, so dass das Programm nicht missbräuchlich verwendet werden kann.

Anforderung: Verteilte Administration

Es muss die Möglichkeit bestehen, dass die Systeme verteilt administriert werden können. Dadurch bleiben neben den Sicherheitsanforderungen bezüglich der Infrastruktur der Interessengruppen auch die Eigentumsrechte an Programmen und Daten beteiligter Interessengruppen gewahrt.

Beispiel: Ein Herstellerbetrieb integriert seine durch aufwendige Versuchsreihen gewonnene Materialdatenbank in eine gemeinsame Wissensbasis. Die verteilte

Administration ermöglicht es dem Eigentümer, eigenverantwortlich zu entscheiden, welche weiteren Benutzer Zugriffsrechte auf seine Datenbank bekommen.

Anforderung: Dienste-Schnittstelle

Eine Dienste-Schnittstelle zu anderen Systemen wird umgesetzt, so dass eine standortunabhängige Verteilung der Teilsysteme einer gemeinsamen Wissensbasis möglich ist. Es können aber auch Teilsysteme einer Organisation, welche nicht in die gemeinsame Wissensbasis integriert sind, die reine Zugriffssteuerung über diese Dienste-Schnittstelle nutzen. Dadurch wird eine systemübergreifende Nutzung der Zugriffssteuerung erreicht.

Beispiel: Ein Datenbanksystem mit Materialdaten aus eigens durchgeführten Versuchen eines Herstellerbetriebes wird nur innerbetrieblich eingesetzt und darf nicht veröffentlicht werden. Durch die Möglichkeit, über eine Dienste-Schnittstelle die Zugriffssteuerung einer gemeinsamen Wissensbasis zu verwenden, kann die Materialdatenbank im Intranet unter Beibehaltung derselben Benutzerverwaltung, welche durch das Rollenmodell abgebildet wird, eingesetzt werden. Sollen später einzelne Objekte der Materialdatenbank externen Benutzern im Rahmen eines Teilsystems der gemeinsamen Wissensbasis zur Verfügung gestellt werden, so ist dies ebenfalls über die Dienste-Schnittstelle möglich. Die Netzwerkinfrastruktur des Herstellerbetriebes bleibt in jedem Fall gewahrt.

Anforderung: Redundanzvermeidung und Mehrfachverwendung

Redundanzvermeidung und Mehrfachverwendung formulieren eine Anforderung, welche indirekt in den vorherigen Anforderungen bereits enthalten ist. Sie werden aber hier noch einmal explizit aufgeführt, da diese beiden Punkte sehr wichtige Aspekte für den Aufbau einer konsistenten gemeinsamen Wissensbasis darstellen.

Durch die Kopplung der Zugriffssteuerung mit Sichten können dieselben Objekte als Teil mehrerer Anwendungen verwendet und in dafür optimierten Sichten dargestellt werden. Dadurch werden Redundanzen vermieden und sowohl die Objekte selbst, als auch die Anwendungen können mehrfach in unterschiedlichen Kontexten verwendet werden.

Beispiele für Möglichkeiten der redundanzfreien Mehrfachverwendung:

- Bestehende Insellösungen werden über eine gemeinsame Schnittstelle integriert und stehen somit der redundanzfreien Mehrfachverwendung zur Verfügung.
- Dieselben Programme (z.B. Simulationsprogramme) können für unterschiedliche Anwendungsfälle im Hintergrund arbeiten. So können bestehende Simulationsprogramme, die beispielsweise im Rahmen eines Forschungsprojektes entstanden sind, mit speziell optimierten Sichten für neue Verwendungszwecke, etwa im Bereich der Wissensvermittlung, verwendet werden.

Vorteile, die sich aus der redundanzfreien Mehrfachverwendung ergeben:

- Einfache Adaptierbarkeit einer bestehenden Funktion für eine neue Rolle oder einen neuen Anwendungsfall.
- Redundanzfreie Wissensspeicherung ist direkt mit einer Erhöhung der Datenkonsistenz verbunden.
- Eine konsistente Benutzerverwaltung über die gesamte Wissensbasis ist möglich.

Basisarchitektur und prototypische Umsetzung

Aufbauend auf das Rollenmodell wird eine Basisarchitektur für Web-Anwendungen entwickelt. Diese bildet die Grundlage für Softwareprojekte im Bereich Kerntechnik, die auf dem Rollenmodell, welches mit der Sichtensteuerung gekoppelt ist, aufbauen. Die Architektur setzt die Anforderungen um und ermöglicht die kontinuierliche Überprüfung der Erweiterungen während der gesamten Modellentwicklung.

1.6 Fallbeispiele

Da die idealtypischen Bereiche des Wissens nicht scharf voneinander getrennt werden können, wird bei den folgenden Fallbeispielen angestrebt, eine Zuweisung im Hinblick auf die für das jeweilige Beispiel relevanten Aspekte vorzunehmen.

1.6.1 Fallbeispiel Simulationsprogramm - Wkind

Ein Beispiel eines Simulationsprogramms ist Wkind [12]. Dieses Programm koppelt die neutronenphysikalischen Prozesse mit der Thermohydraulik in einem Hochtemperaturreaktor (HTR)-Kern. Dadurch können die Transienten eines HTR berechnet

werden. Bei der Entwicklung des Programms fand die Optimierung auf einen spezifischen Reaktortyp statt.

Bisherige Anwendung des Simulationsprogramms

Der Einsatz des Simulationsprogramms beschränkte sich bisher auf die Anwendung durch Experten auf dem Gebiet der Kerntechnik, welche das Programm nach entsprechend großer Einarbeitungszeit für Untersuchungen einsetzen konnten. Die Auswertung der Simulationsergebnisse und deren Visualisierung erfolgten durch manuelle Verwendung entsprechender Werkzeuge.

Welchen Mehrwert bringt die Einführung unterschiedlicher Sichten für die Verwendung des Simulationsprogramms?

Es ist wünschenswert, ein solches Simulationsprogramm in seiner bestehenden Form für unterschiedliche Aufgaben verwenden zu können, wobei der Einarbeitungsaufwand zur Nutzung des Programms für den jeweiligen Anwendungsfall nicht zu hoch sein darf. So ist beispielsweise der Einsatz dieses Simulationsprogramms sowohl für die Untersuchungen eines Experten auf dem Gebiet der Kerntechnik, als auch für die Lehre im Rahmen eines Lehr-/Lernsystems denkbar. Im zweiten Fall allerdings mit einem auf die Aufgabenstellung reduzierten Parameterumfang.

Um die oben beschriebene Mehrfachverwendung eines Simulationsprogramms zu ermöglichen, müssen für den jeweiligen Anwendungsfall und somit für jede Rolle, in der das Simulationsprogramm verwendet werden soll, optimierte Sichten erzeugt werden. Diese müssen für das Simulationsprogramm die jeweils benötigten Funktionen in einer dafür optimierten Ausprägung anbieten.

1.6.2 Fallbeispiel Informationssystem - Sinter Network

Erste Ansätze, ein einheitliches Informationssystem mit verschiedenen Untersystemen im Bereich Kerntechnik einzurichten, ist SINTER¹⁵ Network [13], eine Entwicklung des Instituts für Kernenergetik und Energiesysteme der Universität Stuttgart. Es handelt sich bei SINTER Network um ein geschlossenes Netzwerk, das sich aus verschiedenen lokalen Untersystemen zusammensetzt. Die Benutzer, die im wesentlichen aus Hochschulen, Forschungsorganisationen und Herstellerbetrieben stammen, können SINTER Network als Kooperationswerkzeug verwenden und damit bei-

¹⁵ Sustainable & Innovative Nuclear Technology Evolution – R&D Network

spielsweise Informationen über gemeinsame Forschungsvorhaben austauschen. Es befindet sich eine Sammlung von Berichten in SINTER Network, die mit Zugriffsberechtigungen versehen werden können. Aufbauend auf ein einfaches Benutzermodell wird die Möglichkeit zur Gruppenbildung geschaffen, um dadurch einzelne Bereiche des Systems mit unterschiedlichen Zugriffsberechtigungen zu versehen. Die Gruppen stellen eine Beschränkung für den Informationsaustausch dar, die Neuanlage der Gruppen ist durch die einzelnen Benutzer möglich und wird von ihnen auch selbständig verwaltet.

Bisherige Anwendung des Informationssystems

Die bisherige Hauptanwendung des Informationssystems bestand darin, Forschungsberichte im Rahmen des geschlossenen Benutzernetzwerks in Form einer Online-Bibliothek auszutauschen.

Welchen Mehrwert bringt die Einführung unterschiedlicher Sichten für die Verwendung des Informationssystems?

Durch die Einführung unterschiedlicher Sichten, welche direkt mit den Rollen der Zugriffssteuerung gekoppelt sind, können die Forschungsberichte in eine gemeinsame Wissensbasis integriert werden. Je nach Rolle eines Benutzers können spezielle Sichten eingesetzt werden, welche beispielsweise nur öffentlich freigegebene Teile der Berichte darstellen, die internen Teile bleiben den privilegierten Rollen vorbehalten. Dadurch können die Inhalte in anderen Kontexten und von weiteren berechtigten Interessengruppen (z.B. Regulierungsbehörden, Herstellerbetrieben, Energieversorgungsunternehmen, Hochschulen) verwendet werden, denen bisher ein Zugriff auf alle Teile (öffentlich und intern) des Gesamtberichts hätte verwehrt bleiben müssen. Durch diese Filtermöglichkeiten bleiben die Sicherheit der Wissensbasis und deren Inhalte unter gleichzeitiger Wahrung spezieller Interessen erhalten.

1.6.3 Fallbeispiel Einsatz von ABR-KFUE als Forschungs- und Trainingssystem

Ein weiteres Beispiel aus dem Bereich Kerntechnik sind Simulationsanwendungen für den Notfallschutz. Hierfür werden von den Umweltministerien der Bundesländer Baden-Württemberg und Rheinland-Pfalz Simulationsprogramme, die vom Institut für Kernenergetik und Energiesysteme entwickelt wurden, im Rahmen des System ABR-KFUE [14] eingesetzt. Das System dient der Kernreaktor-Fernüberwachung und

berechnet bei einem eintretenden Störfall selbständig die luftgetragene Ausbreitung der radioaktiven Schadstoffe. Parallel dazu können mit dem System Situationen und Vorgänge simuliert und analysiert werden.

Bisherige Anwendung des ABR-KFUE als Forschungs- und Trainingssystem

Das System ABR-KFUE ist bisher ausschließlich für den Notfallschutz und für Übungen zur Bedienung der Oberfläche des Klienten¹⁶ ausgelegt. Deswegen kann es nur bedingt für Forschungsaufgaben und für weiterführende Übungen des Katastrophenschutzes eingesetzt werden.

Welchen Mehrwert bringt die Einführung unterschiedlicher Sichten für die Verwendung von ABR-KFUE als Forschungs- und Trainingssystem?

Durch die Überführung des Berechnungskerns von ABR-KFUE in ein neues Forschungssystem, das gekoppelte Rollen und Sichten unterstützt, können neben den reinen Forschungsaufgaben auch Übungen im Bereich Notfallschutz besser vorbereitet werden. Diese Mehrfachverwendung wird erst durch die Einführung rollenabhängig gesteuerter Sichten erreicht, da für die typischen Benutzer aus dem Bereich Notfallschutz teilweise andere Sichten und somit ein anderes Ein- und Ausgabeverhalten benötigt werden, als für typische Benutzer aus dem Bereich der Forschung.

Beispiele für Sichten im Bereich Forschung

- Sicht für die Erstellung von Parameterstudien
- Sicht für die Erstellung von Modellanalysen
- Sicht für mobile Emissionsstandorte

Beispiele für Sichten im Bereich Training:

- Sicht für mobile Emissionsstandorte
- Sicht zur Übung der Ergebnisinterpretation und der daraus notwendigen Maßnahmen im Störfall
- Sicht zur Vorbereitung realer Übungen des Katastrophenschutzes

¹⁶ Der ABR-KFUE-Klient ist eine eigenständige Anwendung, welche eine Benutzeroberfläche zur Verwendung des Notfallschutzsystems ABR-KFUE zur Verfügung stellt.

2 Grundlagen

In diesem Kapitel werden die Grundlagen zur Entwicklung des Rollenmodells für den Fachbereich Kerntechnik beschrieben. Wir beginnen mit der Entwicklung der Bewertung der Zugriffssteuerungsmethoden. Danach wird der Stand der Technik im Bereich der Referenzmodelle allgemein und in einem weiteren Unterkapitel unter besonderer Berücksichtigung des genormten Rollenreferenzmodells (Role-Based Access Control) dargestellt. Ausgehend von einer Kritik am bestehenden Modell werden die konkreten Anforderungen des im Rahmen dieser Arbeit entwickelten sichtengesteuerten Rollenmodells (Role-Based View Control) hergeleitet und erläutert. Im letzten Teil des Kapitels wird das bei der Modellentwicklung und Umsetzung verwendete Vorgehensmodell beschrieben.

2.1 *Stand der Technik*

2.1.1 Entwicklung der Bewertung der Zugriffssteuerungsmethoden

Im Folgenden wird die Entwicklung der Standards zur Bewertung von Zugriffssteuerungsmethoden dargestellt. In den Standards sind die Voraussetzungen festgelegt, die ein Softwaresystem erfüllen muss, um von offiziellen Stellen einer Sicherheitsklasse zugeordnet und dafür zertifiziert werden zu können. Diese Methoden wurden zuerst im Standard *Trusted Computer System Evaluation Criteria*¹⁷ (TCSEC) des amerikanischen Verteidigungsministeriums (Department of Defence - DoD) im Jahre 1983 in Schutzstufen eingeteilt [15] und in der 2. Auflage im Jahre 1985 aktualisiert [16]. In Europa wurde im Jahr 1991 von der Europäischen Kommission der Standard *Information Technology Security Evaluation Criteria* (ITSEC) [17] veröffentlicht, der sich stark an dem älteren deutschen Standard *IT-Sicherheitskriterien* (ITSK¹⁸) [18] orientierte. Der neue und heute aktuelle Standard *Common Criteria for Information Technology Security Evaluation* (CC) ist ein internationaler Standard für die gemeinsamen Kriterien zur Bewertung der Sicherheit in der Informationstechnologie. Er soll den amerikanischen Standard TCSEC, den europäischen Standard ITSEC und den

¹⁷ Diese Veröffentlichung wird in der Literatur häufig auch als *Orange Book* bezeichnet.

¹⁸ Die ITSK wurde im Jahre 1989 von der Zentrale für Sicherheit in der Informationstechnologie veröffentlicht, die heute umbenannt wurde in Bundesamt für Sicherheit in der Informationstechnik (BSI). Diese Veröffentlichung ist auch unter dem Namen Grünbuch bekannt.

kanadischen Standard CTCPEC¹⁹ zukünftig ablösen. Der Standard CC wurde im April 1999 in der Version 2.3 [19] zur ISO²⁰-Norm [20]-[22] erklärt, die aktuellste Fassung dieses Standards ist Version 3.1 [23] von September 2006. Diese wurde bei der ISO zur Standardisierung eingereicht [24].

2.1.2 Referenzmodelle für die Zugriffssteuerung

Mandatory Access Control

Mandatory Access Control (MAC) [16] [25] ist ein regelbasiertes Zugriffssteuerungsverfahren, das hauptsächlich im militärischen Bereich und bei Behörden in solchen Bereichen eingesetzt wird, bei denen es überwiegend um die Vertraulichkeit und Konsistenz von Daten ankommt. Die Zugriffsberechtigungen bei MAC werden anhand der Eigenschaften des Subjektes (z.B. Benutzer, Prozess) und des Objektes (z.B. Datei, Verzeichnis, Datenbank) entschieden. Eine Zugriffsentscheidung erfolgt immer nach allgemeinen Regeln. In diesen Regeln ist beispielsweise festgelegt, dass der Zugriff eines Subjektes auf ein Objekt nur dann erfolgen kann, wenn das Subjekt als Eigenschaft die notwendige Freigabestufe besitzt, die es benötigt, um auf Objekte mit Informationen dieser Sensitivitätsstufe zuzugreifen.

Der Zugriff auf ein MAC-geschütztes System wird niemals direkt ausgeführt, sondern geschieht immer über einen Referenzmonitor²¹.

Discretionary Access Control

Discretionary Access Control (DAC) [16] [25] ist ein Zugriffssteuerungsverfahren, das in unterschiedlichen Ausprägungen sehr weite Verbreitung findet.

DAC erlaubt es den Subjekten eines Systems, den Zugriff auf Objekte, die unter ihrer Kontrolle liegen, anderen Subjekten zu gewähren oder zu verwehren. Je nach Ausprägung der Implementierung können die Subjekte nur die Akteure selbst (z.B. Benutzer und Prozesse) oder auch Gruppen dieser Akteure (z.B. Benutzergruppen) umfassen.

¹⁹ *Canadian Trusted Computer Product Evaluation Criteria* ist ein kanadischer Standard aus dem Jahre 1993 und stellt eine Kombination aus den Ansätzen TCSEC und ITSEC dar.

²⁰ *International Organization for Standardization* (ISO)

²¹ Der Referenzmonitor ist in einem Softwaresystem eine logische Einheit. Er entkoppelt sämtliche Zugriffe von Subjekten auf Objekte. Er ist als alleinige Instanz für sämtliche Zugriffe eines Systems verantwortlich.

Heute werden DAC oder davon abgewandelte Zugriffsteuerungssysteme durch den geringeren Verwaltungsaufwand sehr häufig für die Umsetzung der unterschiedlichsten Benutzermodelle in Softwaresystemen eingesetzt. Beispiele sind der Einsatz in den Linux-Betriebssystemen und bei Microsoft Windows.

Role-Based Access Control

Vor der Einführung von *Role-Based Access Control* (RBAC) beschränkten sich die Möglichkeiten der Zugriffssteuerung auf die beiden Modelle MAC und DAC. Dies führte dazu, dass im Allgemeinen davon ausgegangen werden konnte, dass alle Systeme entweder auf dem einen oder dem anderen Modell beruhten. Erst durch die Einführung von RBAC wurde eine Alternative zu den bis dahin bekannten Modellen gefunden, bei dem die Zugriffe nicht mehr anhand allgemeiner Regeln oder der Handlungsfreiheit des Benutzers bestimmt wurden, sondern anhand der Aufgabe eines Benutzers in einer Organisation.

Die erste formale Beschreibung von RBAC stammt aus dem Jahre 1992 von Ferraiollo und Kuhn [26] und wurde dort als die Methode des *non-discretionary access control* vorgestellt. Die dort beschriebenen Rollen haben große Ähnlichkeit mit konventionellen Benutzergruppen, die auch im Bereich DAC vorkommen. Im Gegensatz zu den Gruppen in Abb. 4-b, in denen Benutzer gesammelt werden, bestehen die Rollen in Abb. 4-a aus einer Sammlung von Berechtigungen. Die Zuweisung von Berechtigungen kann bei der Verwendung von Gruppen (Abb. 4-b) sowohl direkt über den Benutzer, als auch indirekt über die Gruppe erfolgen. Die dadurch entstehende Umgehungsmöglichkeit, die zu Sicherheitsproblemen führen kann, wurde bei RBAC (Abb. 4-a) dadurch verhindert, dass jeder Zugriff nur über die Rollen durchgeführt werden kann.

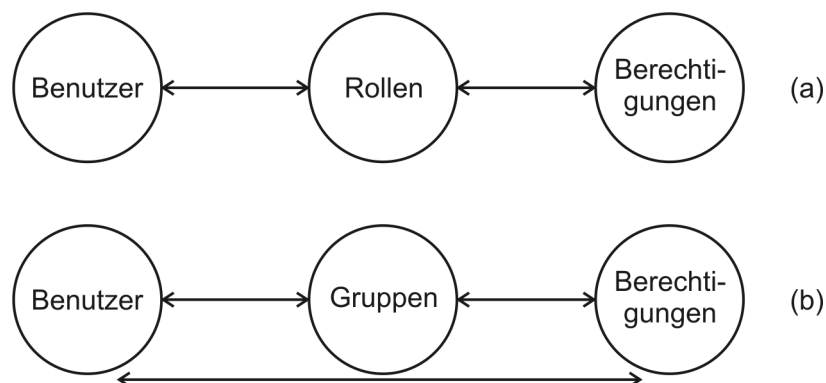


Abb. 4: Zuordnung von Berechtigungen zu Benutzern nach rollen- (RBAC) und gruppenbasierten Modellen [27]

Es gibt eine ANSI-Norm für *Role-Based Access Control* (RBAC) [28], in der die grundlegenden Funktionen für rollenbasierte Zugriffssteuerungssysteme in Form eines allgemeinen RBAC-Referenzmodells (Kap. 2.2) definiert werden. Dieses Referenzmodell bildet somit ein idealtypisches Modell in abstrakter Form ab, auf dessen Basis das spezielle Modell für den Fachbereich Kerntechnik entwickelt wird.

2.2 RBAC-Referenzmodell

2.2.1 Begriffsdefinitionen

Alle Abbildungen, die sich im Folgenden auf das genormte RBAC-Referenzmodell oder auf direkte Erweiterungen dessen beziehen, verwenden konsequent die in den Klammern aufgeführten englischen Begriffe und Abkürzungen der ANSI-Norm. Im Fließtext werden die deutschen Begriffe verwendet. Werden Abbildungen beschrieben, so werden die englischen Begriffe und Abkürzungen in Klammern angegeben. Die im Rahmen dieser Arbeit neu eingeführten Begriffe des erweiterten RBAC-Referenzmodells werden analog dazu in den Abbildungen ebenfalls in Englisch ausgeführt, um diese konsistent zu halten.

Die folgenden Begriffsdefinitionen führen die Basis-Elemente des RBAC-Referenzmodells ein:

- **Benutzer (USERS):** Ein Benutzer wird repräsentiert durch eine Person oder eine Schnittstelle zu einem anderen Programm (z.B. über einen Web-Service).
- **Rollen (ROLES):** Eine Rolle ist eine Aufgabe²² im Kontext einer Organisation mit einer zugehörigen Semantik bezüglich der verliehenen Befugnis und Verantwortung, die ein Benutzer erhält, wenn er der Rolle zugewiesen wird.
- **Sitzungen (SESSIONS):** Eine Sitzung ist eine Instanz eines Benutzerdialogs. Der Begriff ist gleichbedeutend mit dem traditionell verwendeten Begriff Subjekt (subject), der früher für einen Prozess, der im Namen eines Benutzers ausgeführt wird, verwendet wurde.

²² Im Rahmen dieser Arbeit wird für den Begriff *Funktion im Kontext einer Organisation* stattdessen der Begriff *Aufgabe im Kontext einer Organisation* verwendet. Dies liegt daran, dass das Rollenmodell um das Element *Funktion* erweitert wurde, das im Gegensatz zur *Funktion im Kontext einer Organisation* eine *Funktion im Rahmen eines Softwaresystems* meint.

- **Objekte (OBS):** Ein Objekt ist jede beliebige Ressource, die der Zugriffssteuerung unterliegt, und auf die von einem Benutzer im Rahmen einer Sitzung zugegriffen wird.
- **Operationen (OPS):** Eine Operation ist eine ausführbare Methode, die im Rahmen einer Sitzung aufgerufen wird.
- **Berechtigungen (PRMS):** Eine Berechtigung wird benötigt, um eine Operation auf ein Objekt auszuführen.

2.2.2 Basis-RBAC-Referenzmodell

Das Basis-RBAC-Referenzmodell (Core RBAC Model) in Abb. 5, das als ANSI-Norm [28] herausgegeben wurde, besteht in dieser einfachen Variante aus dem zentralen Element Rollen (ROLES), die jeweils Beziehungen der Kardinalität m:n zu den angeschlossenen Elementen Benutzer (USERS), Sitzungen (SESSIONS) und den Berechtigungen (PRMS) haben. Die Berechtigungen selbst bestehen aus Operationen (OPS) und Objekten (OBS).

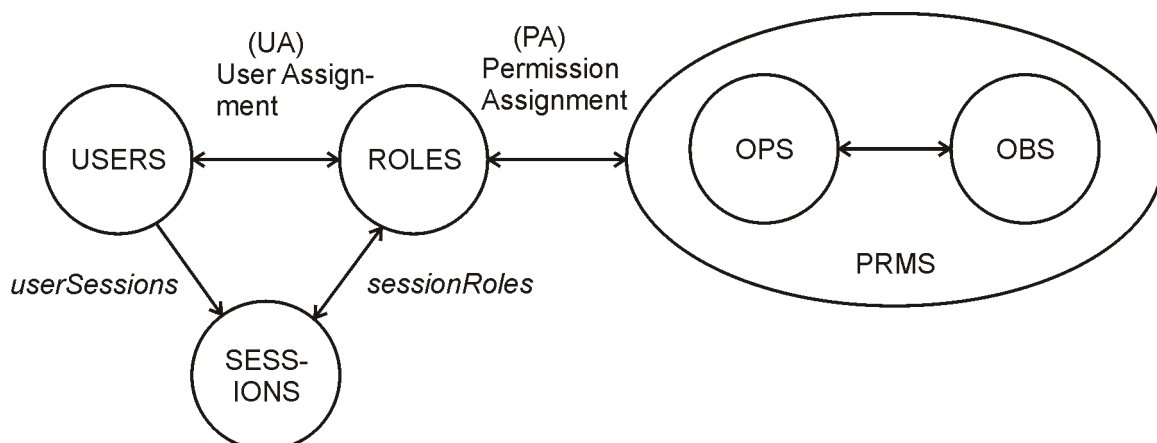


Abb. 5: Core RBAC Model nach ANSI INCITS 359-2004 [28]

Die Relation *User Assignment* (UA) dient der Zuweisung von Benutzern zu den Rollen. Die Rollen beschreiben in abstrakter Form die Aufgaben in einer Organisation. Durch die Zuweisung von Benutzern können diese in der jeweiligen Rolle im System aktiv werden.

Die Relation *Permission Assignment* (PA) weist den Rollen als alleiniger Instanz des RBAC-Modells die Berechtigungen zum Zugriff auf die Objekte zu. Jede Berechtigung setzt sich aus einer Menge von Operationen auf Objekte zusammen.

Durch die Entkopplung der Berechtigungen von den Benutzern können Änderungen bei den Aufgaben eines Benutzers ohne entsprechende Änderungen der Berechtigungen vorgenommen werden. Es müssen nur die entsprechenden Rollenzuweisungen des Benutzers geändert werden.

Nach erfolgreicher Authentifizierung wird für jeden Benutzer eine Sitzung angelegt, welche über die gesamte Laufzeit bis zur Beendigung der Sitzung bestehen bleibt. Hat ein Benutzer z.B. auf eine Web-Anwendung gleichzeitig mehrere Zugriffe, beispielsweise von unterschiedlichen Rechnern, so wird für jeden Zugriff eine eigene Sitzung angelegt. Innerhalb einer Sitzung eines Benutzers werden je nach gewünschter Operation auf die Objekte einzelne Rollen aktiviert und deaktiviert.

In den folgenden Unterkapiteln werden die Beziehungen der zentralen Elemente des Basis-RBAC-Referenzmodells als Grundlage für die weiteren Betrachtungen formal beschrieben.

Rollen - Benutzer (UA – User Assignment)

Die Beziehung UA mit der Kardinalität m:n bildet die Zugehörigkeit der Benutzer (USERS) zu den Rollen (ROLES) ab. UA ist eine echte Teilmenge der Produktmenge aus den Benutzern mit den Rollen. Sie ist spezifiziert als

$$UA \subseteq USERS \times ROLES .$$

Für die Abbildung der Rollen (ROLES) auf eine Menge von Benutzern (USERS)

$$assignedUsers(r : ROLES) \rightarrow 2^{USERS}$$

sieht die ANSI-Norm als formale Spezifikation vor:

$$assignedUsers(r) = \{u \in USERS | (u, r) \in UA\}$$

Rollen - Berechtigungen (PA – Permission Assignment)

Die Relation PA mit der Kardinalität m:n bildet die Beziehung der Rollen (ROLES) zu deren Berechtigungen (PRMS) ab. PA ist eine echte Teilmenge der Produktmenge aus den Berechtigungen mit den Rollen. Sie ist spezifiziert als

$$PA \subseteq PRMS \times ROLES$$

Für die Abbildung der Rollen (ROLES) auf eine Menge von Berechtigungen (PRMS)

$$\text{assignedPermissions}(r : \text{ROLES}) \rightarrow 2^{\text{PRMS}}$$

sieht die ANSI-Norm als formale Spezifikation vor:

$$\text{assignedPermissions}(r) = \{p \in \text{PRMS} \mid (p, r) \in \text{PA}\}$$

Operation – Objekte – Berechtigungen

Die Abbildung zwischen der Menge der Operationen (OPS), die den Berechtigungen (PRMS) zugewiesen sind, ergeben sich laut ANSI-Norm wie folgt:

$$\text{Op}(p : \text{PRMS}) \rightarrow \{op \subseteq \text{OPS}\}$$

Umgekehrt gilt für die Abbildung zwischen der Menge der Objekte (OBS), die den Berechtigungen (PRMS) zugewiesen sind, nach ANSI-Norm folgende Beziehung:

$$\text{Ob}(p : \text{PRMS}) \rightarrow \{ob \subseteq \text{OBS}\}$$

Benutzer – Sitzungen (sessionUsers)

Es wird definiert, dass SESSIONS eine Menge an Sitzungen darstellt. Im Basis-RBAC-Referenzmodell ergibt sich für die Abbildung der Sitzungen auf die Benutzer (USERS) die Beziehung

$$\text{sessionUser}(s : \text{SESSIONS}) \rightarrow \text{USERS}$$

mit der Kardinalität n:1.

Rollen - Sitzungen (sessionRoles)

Die Relation sessionRoles mit der Kardinalität m:n bildet die Beziehung der Sitzungen (SESSIONS) zu deren Rollen (ROLES) ab. Für die Menge der Rollen in Abhängigkeit von der Sitzung

$$\text{sessionRoles}(s : \text{SESSIONS}) \rightarrow 2^{\text{ROLES}}$$

sieht die ANSI-Norm als formale Spezifikation vor:

$$\text{sessionRoles}(s_i) \subseteq \{r \in \text{ROLES} \mid (\text{sessionUser}(s_i), r) \in \text{UA}\}.$$

Die Abbildung der Sitzungen auf die Menge der verfügbaren Berechtigungen

$$availSessionPerms(s : SESSIONS) \rightarrow 2^{PRMS},$$

die einem Benutzer zur Verfügung stehen, sind formal definiert als

$$availSessionPerms(s) = \bigcup_{r \in sessionRoles(s)} assignedPermissions(r).$$

2.2.3 Hierarchisches RBAC

Der ANSI-Standard für RBAC sieht als mögliche Erweiterung zwei Hierarchiearten für RBAC vor. Diese beiden Arten werden als *General Role Hierarchies* und *Limited Role Hierarchies* bezeichnet und sind in Abb. 6 dargestellt.

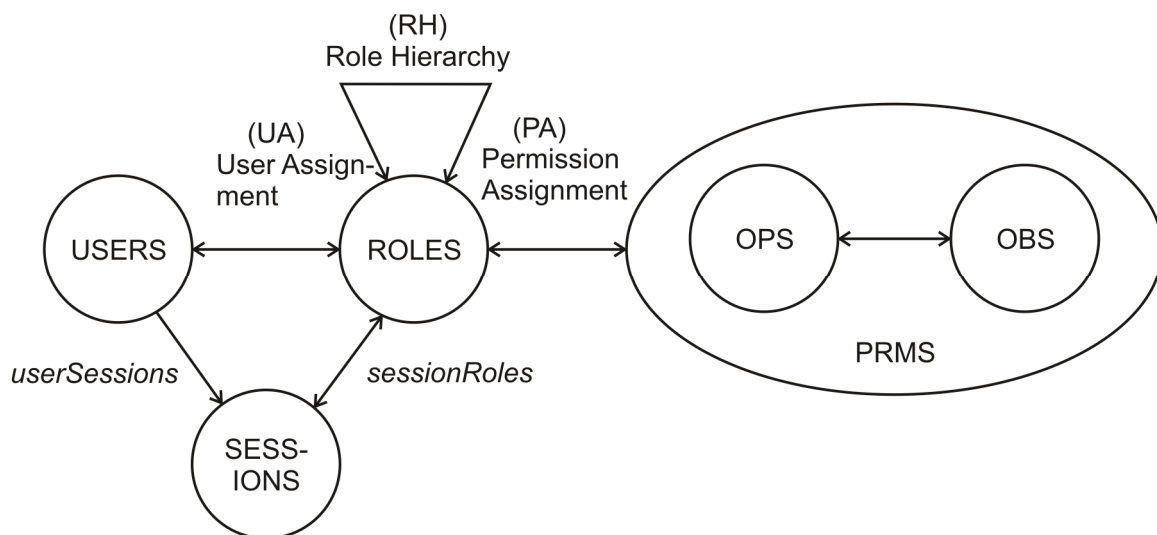


Abb. 6: Rollenhierarchien [28]

Rollenhierarchien (Role Hierarchy – RH) beschreiben eine Vererbungsbeziehung zwischen Rollen. Wenn beispielsweise die Rolle r_1 der Nachfolger von der Rolle r_2 ist, erbt sie nach der ANSI-Norm alle Privilegien des Vorgängers. In der Norm beziehen sich diese Privilegien nur auf die Berechtigungen, während in dem erweiterten Modell, das im Rahmen dieser Arbeit vorgestellt wird, auch Sichten vererbt werden können (vgl. Kap. 3.4.1). Eine Vererbung der Benutzer findet jedoch nicht statt.

Darstellung der Rollenhierarchien in Diagrammen

Hierarchien können mit Hilfe von erweiterten Hasse-Diagrammen [29] visualisiert werden. Zur Vereinheitlichung der Darstellung werden die Rollen als Kreise dargestellt. Die Benutzer, welche den Rollen zugewiesen werden, werden als doppelte Kreise ausgeführt. Die Vererbungsrichtung von den Vorgängern zu den Nachfolgern

ist definiert von unten nach oben. In Abb. 7 ist die Vererbungsbeziehung zwischen Nachfolgern, deren Bezeichnungen mit dem Anfangsbuchstaben D beginnen, und dem Vorgänger, dessen Bezeichnung mit dem Anfangsbuchstaben A beginnt, eingezeichnet. Die Pfeilspitze zeigt immer in Richtung des Vorgängers, dessen Berechtigungen geerbt werden. Die Zuweisung eines Benutzers, dessen Bezeichnung mit dem Anfangsbuchstaben U beginnt, zu einer Rolle wird ebenfalls mit einem Pfeil zwischen dem Benutzer und der Rolle dargestellt. Die Pfeilspitze zeigt in Richtung der Rolle, deren Berechtigungen ein Benutzer erhält.

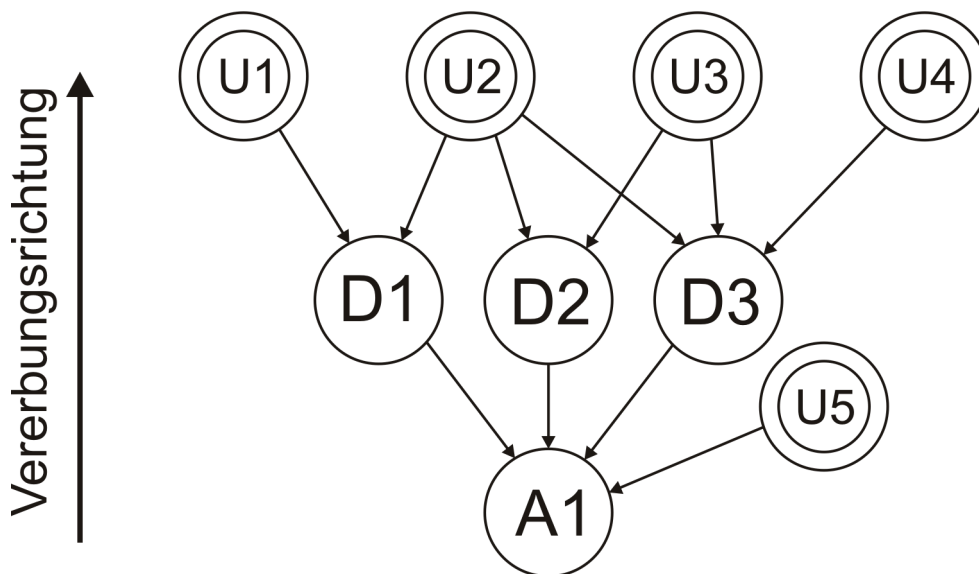


Abb. 7: Beispiel eines Hasse-Diagramms mit Rollen, Vererbung und Benutzern

Generelle Rollenhierarchien - General Role Hierarchies

Zur Darstellung einer unmittelbaren Vererbungsbeziehung zwischen zwei Rollen, wenn also keine andere Rolle in der Hierarchie zwischen dem Vorgänger und dem Nachfolger liegt, dann wird das Größergleichzeichen \geq verwendet. Die Rollenvererbung $r_1 \geq r_2$ bedeutet, dass der Nachfolger r_1 alle Berechtigungen vom Vorgänger r_2 erbt. Die m:n-Beziehung RH aus Abb. 6 ist eine Halbordnung der Produktmenge der Rollen. Sie ist spezifiziert als:

$$RH \subseteq ROLES \times ROLES$$

Die formale Spezifikation der autorisierten Berechtigungen der Vorgängerrolle r_2 ist eine echte Teilmenge der autorisierten Berechtigungen der Nachfolgerrolle r_1 . Formal wird die Beziehung spezifiziert als:

$$r_1 \geq r_2 \Rightarrow \text{authorizedPermissions}(r_2) \subseteq \text{authorizedPermissions}(r_1) \wedge \\ \text{authorizedUsers}(r_1) \subseteq \text{authorizedUsers}(r_2)$$

Die Abbildung der Rollen auf eine Menge von Benutzern

$$\text{authorizedUsers}(r : \text{ROLES}) \rightarrow 2^{\text{USERS}},$$

die autorisiert sind, wird formal definiert als

$$\text{authorizedUsers}(r) = \{u \in \text{USERS} \mid r' \geq r, (u, r') \in \text{UA}\}.$$

Die Abbildung der Rollen auf eine Menge von Berechtigungen

$$\text{authorizedPermissions}(r : \text{ROLES}) \rightarrow 2^{\text{PRMS}}$$

wird für die autorisierten Berechtigungen formal spezifiziert als

$$\text{authorizedPermissions}(r) = \{p \in \text{PRMS} \mid r' \geq r, (p, r') \in \text{PA}\}$$

Eingeschränkte Rollenhierarchien - Limited Role Hierarchies

Diese eingeschränkten Rollenhierarchien werden an dieser Stelle der Vollständigkeit wegen aufgeführt, da hierfür eine eigene Umsetzung gewählt wurde (vgl. Kap. 3.3.3). Die Norm sieht vor, dass jede Rolle entweder keinen oder exakt einen Vorgänger haben kann. Für alle Rollen r , r_1 und r_2 gilt, dass wenn r Nachfolger von r_1 ist und r Nachfolger von r_2 ist, so müssen r_1 und r_2 die gleiche Vorgängerrolle sein. Formal wird dies definiert durch:

$$\forall r, r_1, r_2 \in \text{ROLES}, r \geq r_1 \wedge r \geq r_2 \Rightarrow r_1 = r_2.$$

2.2.4 Aufgabenteilung

Die ANSI-Norm des RBAC-Modells sieht optional die Aufgabenteilung (Separation of Duty – SoD) vor und unterteilt diese in zwei Bereiche:

1. Statische Aufgabenteilung
2. Dynamische Aufgabenteilung

Aufgabenteilung findet dann Einsatz, wenn man eine Möglichkeit vorsehen möchte, Rollen generell oder bedingt durch den bisherigen zeitlichen Ablauf des Systems gegenseitig auszuschließen.

Statische Aufgabenteilung – Static Separation of Duty

Unter statischer Aufgabenteilung (Static Separation of Duty – SSD) [30] [31] versteht man eine Erweiterung des Basis-RBAC-Referenzmodells, in der sich Rollen statisch gegenseitig ausschließen können.

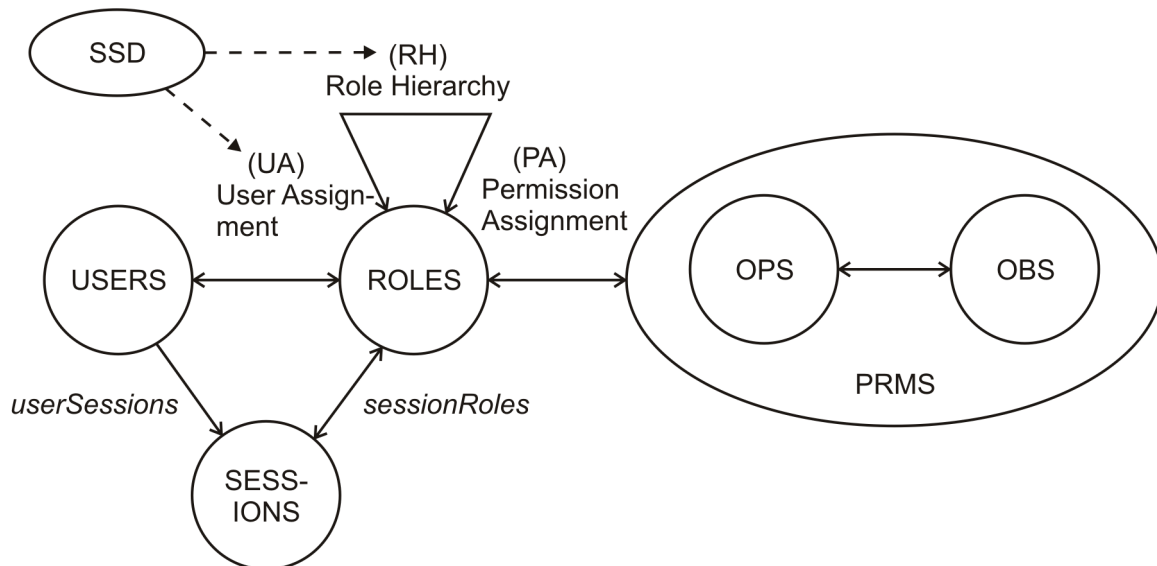


Abb. 8: Statische Aufgabenteilung – Static Separaton of Duty [28]

In Abb. 8 ist dies gekennzeichnet durch den Einfluss von SSD auf die Beziehung UA, in der es um die Zuweisungen der Benutzer zu den Rollen, und auf die Beziehung RH, in der es um die Umsetzung der Rollenhierarchien geht. Ziel einer statischen Aufgabenteilung ist es, zu verhindern, dass zwei Rollen, die sich aufgrund ihrer Aufgabe in einer Organisation gegenseitig ausschließen, gleichzeitig einem Benutzer zugewiesen werden können.

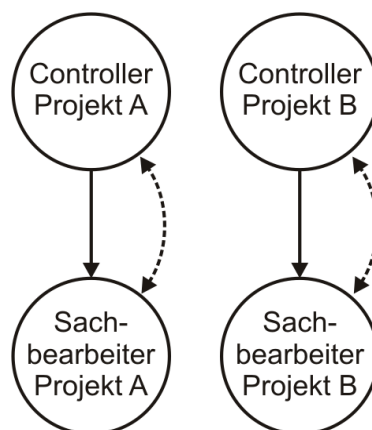


Abb. 9: Beispiel einer statischen Aufgabenteilung

Ein Beispiel für eine statische Aufgabenteilung (Abb. 9) könnten die Rollen Controller und Sachbearbeiter in einem Projekt sein, wenn der Controller Leistungen des Sach-

bearbeiters überprüfen muss. Der gegenseitige Ausschluss zweier Rollen ist im Hasse-Diagramm durch die Verbindung mit einem gestrichelten Pfeil dargestellt. Durch SSD kann in diesem Beispiel verhindert werden, dass ein Benutzer gleichzeitig die Rollen Sachbearbeiter in Projekt A und gleichzeitig Controller in Projekt A ist. Andererseits ist es kein Problem, dass ein Benutzer Controller für ein Projekt wird, in dem er nicht selbst die Rolle des Sachbearbeiters hat.

Die statische Aufgabenteilung (SSD) in Abb. 8 ist definiert als eine echte Teilmenge der Produktmenge aus einer Menge von Rollen und einer natürlichen Zahl, die größer gleich 2 ist:

$$SSD \subseteq (2^{ROLES} \times N).$$

Für alle Paare dieser Produktmenge

$$(rs, n) \in SSD$$

gilt, dass es keine echte Teilmenge t der Menge der Rollen rs geben darf, bei denen ein Benutzer n oder mehr der Rollen von rs zugewiesen bekommen hat. Formal ist dies definiert durch

$$\forall (rs, n) \in SSD, \forall t \subseteq rs : |t| \geq n \Rightarrow \bigcap_{r \in t} authorizedUsers(r) = \{ \}$$

Dynamische Aufgabenteilung – Dynamic Separation of Duty

Dynamische Aufgabenteilung (Dynamic Separation of Duty – DSD) kann in Systemen eingesetzt werden, bei denen Interessenkonflikte entstehen können, wenn ein Benutzer aufgrund der ihm zugewiesenen Rollen bestimmte sicherheitskritische Operationen sequentiell durchführen kann, ohne dass ein weiterer Benutzer dem zustimmen muss. Um einen solchen Interessenkonflikt zu verhindern, wurde die dynamische Aufgabenteilung eingeführt. Es wird angenommen, ein Benutzer besitze zwei Rollen, die sich nicht generell statisch ausschließen. Die Rolle A ist berechtigt, einen bestimmten Vorgang zu initiieren, die Rolle B ist berechtigt, den durch die Rolle A initiierten Vorgang zu genehmigen. Wenn für den gesamten Vorgang das Vier-Augen-Prinzip angewandt werden soll, wird ein Mechanismus benötigt, der verhindert, dass ein Benutzer im Rahmen seiner beliebig vielen Sitzungen eine Umgehung einleiten kann, indem er zuerst in der Rolle A einen Vorgang initiiert und danach in der Rolle B diesen Vorgang genehmigt.

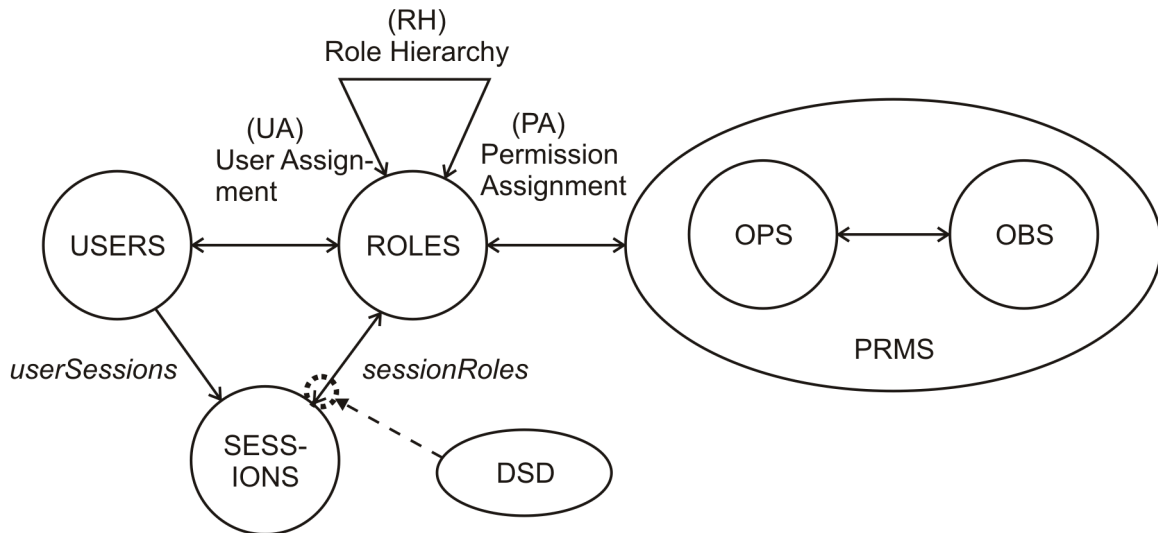


Abb. 10: Dynamische Aufgabenteilung – Dynamic Separation of Duty [28]

In Abb. 10 ist DSD gekennzeichnet durch den Einfluss auf die Beziehung der Sitzungen zu den Rollen. Sie ist definiert als eine echte Teilmenge der Produktmenge aus einer Menge von Rollen und einer natürlichen Zahl, die größer gleich 2 ist.

$$DSD \subseteq (2^{ROLES} \times N)$$

Für alle Paare dieser Produktmenge

$$(rs, n) \in DSD$$

gilt, dass es keine Sitzung geben darf, in der mehr als n der Rollen der Menge rs aktiviert werden dürfen. Formal ist dies definiert durch

$$\forall rs \in 2^{ROLES}, n \in N, (rs, n) \in DSD \Rightarrow n \geq 2, |rs| \geq n,$$

und für

$$\forall s \in SESSIONS, \forall rs \in 2^{ROLES}, \forall roleSubset \in 2^{ROLES}, \forall n \in N, (rs, n) \in DSD$$

gilt

$$roleSubset \subseteq rs, roleSubset \subseteq sessionRoles(s) \Rightarrow |roleSubset| < n.$$

2.2.5 Paketauswahl bei RBAC

Das genormte RBAC-Referenzmodell sieht vor, dass neben der Basis-Variante *Core RBAC* die in Abb. 11 dargestellten Erweiterungen optional ausgewählt werden

können. Dadurch lassen sich je nach Anwendungsfall, Komplexität und Umfang der zu entwickelnden Anwendung die benötigten Pakete optimiert zusammenstellen.

Für das rollenbasierte Benutzermodell des Fachbereichs Kerntechnik, das im weiteren Verlauf dieses Kapitels herausgearbeitet wird, werden zwingend die Pakete Core RBAC und Hierarchische RBAC (Hier. RBAC) in der Variante b (General) benötigt. Die Pakete SSD Relations und DSD Relations können optional je nach Bedarf der Anwendung hinzugenommen oder weggelassen werden.

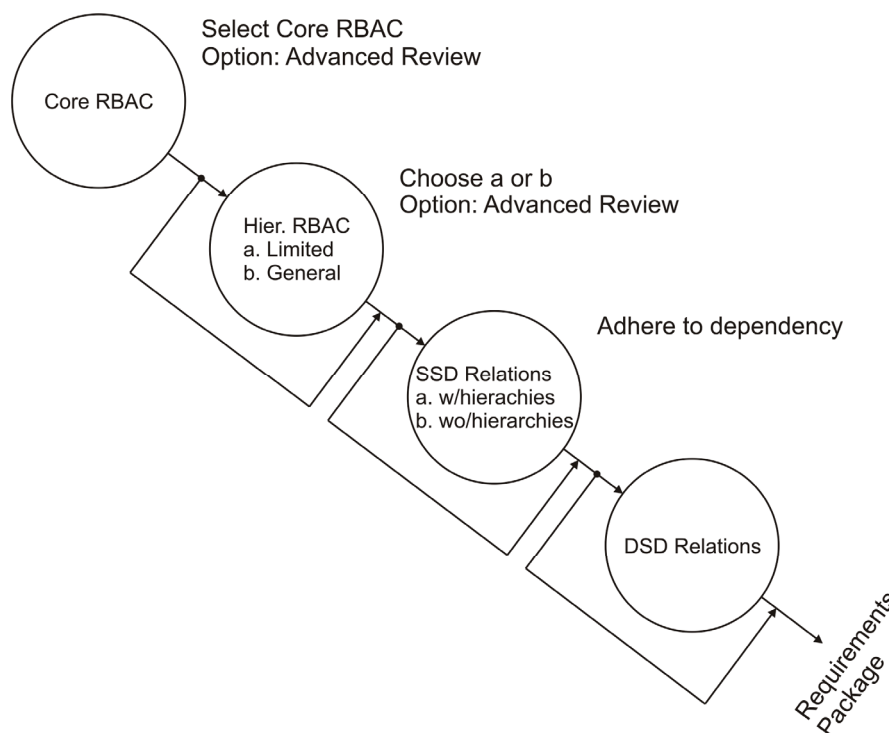


Abb. 11: Methode um funktionale RBAC-Pakete zusammenzustellen [28]

2.3 Kritik am bestehenden RBAC-Referenzmodell

Kritikpunkt 1: Verteilte Administration

Das RBAC-Referenzmodell ist verglichen mit den anderen Referenzmodellen MAC und DAC (Kap. 2.1.2) ein sehr flexibles Modell, das den notwendigen Administrationsaufwand in Grenzen hält. Allerdings sieht es keine Methoden zur verteilten Administration und damit verbunden einer Aufteilung in unabhängige Teilbereiche vor. Integrationssysteme im Bereich Kerntechnik benötigen aber genau hier Lösungen, da neben einem hohen Administrationsaufwand auch die Infrastrukturen der teilnehmenden Interessengruppen nicht gefährdet werden dürfen. Der Aufbau einer gemeinsamen kerntechnischen Wissensbasis hängt somit sehr stark davon ab, dass jede teilnehmende Organisation eigenverantwortlich ihren Bereich verwalten kann.

Beispiel: Datenbank mit vielen zu aktualisierenden Datensätzen

Wenn in einer Datenbank viele Datensätze von unterschiedlichen Organisationen gesammelt und regelmäßig aktualisiert werden, dann ist dies bei einer zentralen Dateneingabe sehr aufwendig. Sinnvoller ist es, logische Einheiten zu bilden und diese von den jeweiligen Interessengruppen eigenständig administrieren zu lassen. Wird nur die Dateneingabe, nicht aber die Administration dezentralisiert, besitzt dies den Nachteil, dass Änderungen in der Mitarbeiterstruktur durch Fluktuation der Verantwortlichkeiten in den Organisationen meist nicht übergreifend und konsistent weitergereicht werden.

Kritikpunkt 2: Objektgranularität, Zugriff und Filterung

Das RBAC-Referenzmodell dient ausschließlich der Steuerung des Zugriffs auf geschützte Objekte. Es macht keine Aussage darüber, wie die darin gespeicherten Informationen dargestellt, innerhalb einer Benutzerschnittstelle bedient, oder nach gewissen Kriterien gefiltert oder aufbereitet werden. Diese Art der Zugriffssteuerung ist immer dann sinnvoll, wenn einerseits wenige Operationen auf viele im Verhalten ähnliche Objekte ausgeführt werden können (z.B. in einem Dateisystem eines Rechners: lesen, schreiben, auflisten, etc.) und andererseits die zu verwaltenden Objekte feingranular und nicht komplex sind. Das existierende Referenzmodell stößt an seine Grenzen, wenn die zu verwaltenden Objekte und Strukturen grobgranular und komplex sind, wie dies im untersuchten Bereich überwiegend der Fall ist.

Beispiel 1: Datenbanken

Datenbanken aus dem Bereich der Systeme für gespeichertes und abrufbares Wissen können komplexe Datensätze enthalten, die grobgranular über einen eindeutigen Identifier angesprochen werden. Intern bestehen diese Datensätze häufig aus einer Vielzahl von Relationen und bilden eine komplexe Hierarchie ab. Werden die Datensätze in unterschiedlichen Kontexten eingesetzt, erlaubt erst der konkrete Anwendungsfall innerhalb einer gemeinsamen Wissensbasis eine Aussage über mögliche Einschränkungen, Filterungen und Optimierungen zu machen. Diese hängen aber immer direkt von der Rolle des Benutzers ab, in der eine Operation auf ein Objekt ausgeführt wird.

Beispiel 2: Simulationsprogramme

Eine Integration von Simulationsprogrammen in eine gemeinsame Wissensbasis und eine daraus resultierende Mehrfachverwendung in unterschiedlichen Kontexten setzt das Vorhandensein jeweils anwendungsspezifischer, optimierter Sichten voraus. Wird beispielsweise dasselbe Simulationsprogramm zur Untersuchung dynamischer Vorgänge in mehreren Anwendungen eingesetzt, wird hierfür mehr als eine reine Zugriffssteuerung, wie sie das RBAC-Referenzmodell bietet, benötigt. Vielmehr ist die reine Zugriffssteuerung auf die grobgranularen Objekte einer Simulationsanwendung nur unzureichend anwendbar, da hierbei keine Aussagen über den Kontext der Verwendung gemacht werden kann. Eine sinnvolle Mehrfachverwendung in unterschiedlichen Anwendungen kann nur dadurch erreicht werden, dass abhängig von der jeweiligen Rolle entsprechende Sichten erstellt und vom Rollenmodell gesteuert werden.

2.4 Erweiterungen durch die Kopplung der Sichtensteuerung mit dem Rollenmodell

In diesem Unterkapitel werden die Erweiterungen, die sich aus der Kopplung des Referenzrollenmodells mit der Sichtensteuerung in Bezug auf die Anforderungen aus Kapitel 1.5 ergeben, aufgezeigt. Dabei werden die Schlüsse aus den Kritikpunkten in Kapitel 2.3 berücksichtigt. Der Fokus des resultierenden Modells liegt nicht mehr auf der Zugriffssteuerung, sondern im Bereich der Sichtensteuerung als deren direkte Erweiterung. Die Anforderung Sicherheit ergibt sich implizit aus den beiden folgenden Erweiterungen und deren Kopplung mit dem zugrundeliegenden RBAC-Referenzmodell.

Erweiterung: Kopplung der Rollen - und Sichtensteuerung

Diese Erweiterung setzt die gleichnamige Anforderung als Basis des neuen Modells um. Sie ermöglicht es, den Zugriff auf die grobgranularen Objekte rollenspezifisch zu optimieren, da nicht nur die Berechtigungen, sondern auch die Ein- und Ausgabe-schnittstellen direkt aus der jeweiligen Rolle der Benutzer abgeleitet werden können. Sie erweitert das Referenzmodell aber auch dahingehend, dass eine Lösung geschaffen wird, gezielt Operationen nur auf bestimmte Objekte ausführen zu können, da innerhalb eines komplexen Systems nicht alle Operationen auf alle Objekte ausgeführt werden dürfen.

Erweiterung: Verteilte Administration

Diese Erweiterung setzt die gleichnamige Anforderung um. Sie ermöglicht es, die Verwaltung der Zugriffsberechtigungen auf die grobgranularen Objekte logisch getrennt vorzunehmen. Dazu werden Bereiche bestehend aus logischen Einheiten geschaffen, die unabhängig voneinander administriert werden können. Die Verantwortlichkeit für eine Einheit wird von einer speziellen Verwalterrolle übernommen, die eine Delegation ihrer Verantwortlichkeit an Untereinheiten vornehmen kann.

2.5 Vorgehensmodell zur Entwicklung des Rollenmodells

Das im Rahmen dieser Arbeit entwickelte Rollenmodell wurde in einem iterativ-inkrementellen Prozess entwickelt. In jeder Iterationsstufe wurde eine Verfeinerung und Erweiterung des Modells vorgenommen. Um die Tragfähigkeit des Modells überprüfen zu können, wurden die einzelnen Iterationsstufen jeweils im Rahmen von Softwareprojekten umgesetzt.

Während des gesamten Entwicklungsprozesses wurden Methoden, die aus dem Bereich der agilen Softwareentwicklung [32] [33] stammen, angewandt. Die Methode der testgetriebenen Entwicklung bietet den Vorteil, dass jede Implementierung immer mit entsprechenden Tests überprüft werden kann. Die Methode des ständigen Refactorings verbessert die Code-Qualität, indem die Wartbarkeit und Erweiterbarkeit der Implementierungen verbessert wird.

Im folgenden Kapitel werden anhand einer Definition der Einsatzszenarien die Zielanforderungen dieser Arbeit konkretisiert und daraus anschließend eine Liste der Funktionsanforderungen abgeleitet. Schließlich wird die Überführung des Referenzmodells in das neue Modell vorgestellt. Parallel zur Modellentwicklung wird das Konzept einer Basisarchitektur entwickelt, das in Kapitel 4 beschrieben wird.

3 Rollenmodell für den Fachbereich Kerntechnik

3.1 Szenarien für den Einsatz eines Rollenmodells

Aus einer Anforderungsanalyse unter besonderer Berücksichtigung der Fallbeispiele aus Kapitel 1.6 werden Einsatzszenarien abgeleitet, in denen das zu entwickelnde Rollenmodell für den Fachbereich Kerntechnik zukünftig eingesetzt werden soll. Allen Szenarien gemeinsam sind die sehr hohen Anforderungen an die Sicherheit, weswegen dieser Punkt bei der Szenarienbeschreibung als obligatorisch vorausgesetzt wird.

3.1.1 Szenario: Simulationsplattform für Forschung und Lehre

Dieses Szenario beschreibt den in der Kerntechnik häufig auftretenden Fall eines Simulationsprogramms, dessen Eingabe- und Ausgabedatenschnittstellen aus Textdateien oder anderen Dateitypen bestehen. Es umfasst auch eine Kopplung mehrerer Simulationsprogramme, die über eine gemeinsame Ablaufsteuerung verknüpft werden können.

In Kapitel 1.3 wurde der erhöhte Einarbeitungsaufwand für die direkte Verwendung der Simulationsprogramme beschrieben. Das Rollenmodell übernimmt in diesem Szenario eine Sichtensteuerung für das Simulationsprogramm, so dass es in unterschiedlichen Ausprägungen mehrfach für Forschung und Lehre verwendet werden kann. Die Sichten stellen eine optimierte Benutzerein- und -ausgabeschnittstelle zur Verfügung, die dem Kenntnisstand eines typischen Benutzers in der anwendungsabhängigen Rolle entspricht.

3.1.2 Szenario: Integrationssystem

Dieses Szenario beschreibt die Integration des fragmentierten kerntechnischen Wissens aus FP-5 (vgl. Kapitel 1), das in Form von Datenbanken und Anwendungen über unterschiedliche Organisationen verteilt ist. Die Quellen sind räumlich getrennt und über das Internet verbunden. Sie unterliegen den Sicherheitsbestimmungen und der Netzwerkintegrität der jeweiligen Organisation.

Das Rollenmodell übernimmt in diesem Szenario die Zugriffssteuerung auf die integrierten unterschiedlichen Daten- und Anwendungsarten.

3.1.3 Szenario: Fernauthentifizierung und -autorisierung

In diesem Szenario soll das Rollenmodell durch ein räumlich getrenntes und über das Internet verbundenes System einer anderen Organisation zur Authentifizierung und Autorisierung verwendet werden können. Dieser Fall tritt ein, wenn beispielsweise aufgrund begrenzter Mittel in Forschungsprojekten keine vollständige Integration eines Systems in ein Basissystem möglich ist, die Zugriffssteuerung aber trotzdem zentralisiert werden soll.

3.1.4 Szenario: Kerntechnisches Informationssystem

Dieses Szenario beschreibt ein kerntechnisches Informationssystem, das unterschiedliche Daten (Kursdaten in der Kerntechnik, Daten über kerntechnische Einrichtungen, etc.) speichern kann.

Weiter wird davon ausgegangen, dass kerntechnische Forschungsberichte oder Konferenzen mit diesem kerntechnischen Informationssystem verwaltet werden können. Das Rollenmodell steuert die Zugriffe und bietet optimierte Sichten, beispielsweise für das Einstellen eines Berichts und dessen Qualitätskontrolle an. Die unterschiedlichen Bereiche des Systems lassen sich getrennt administrieren, da jeweils getrennte Einheiten für Informationen in einem Bereich zuständig sind.

3.1.5 Szenario: Kombinationssystem – gemeinsame kerntechnische Wissensbasis

Dieses Szenario ergibt sich aus den Zielen des FP-6 (vgl. Kapitel 1), eine gemeinsame kerntechnische Wissensbasis zu schaffen. Sie muss in der Lage sein, Datenbanken und Anwendungen unterschiedlicher Organisationen als Integrationssystem einzubinden und diese mit einer Simulationsplattform und mit einem kerntechnischen Informationssystem zu kombinieren. Als Ergebnis entsteht ein System, das als Wissensbasis viele neue Kombinationsmöglichkeiten der unterschiedlichen Quellen für Forschung und Lehre zulässt.

Über das Rollenmodell wird ein einheitlicher Zugang zum Kombinationssystem geschaffen. Durch spezielle Sichten werden unterschiedliche Rollen nach Anwendungsfällen optimal unterstützt.

3.2 Funktionsanforderungen des Rollenmodells

Aus den Einsatzszenarien werden die Funktionsanforderungen für die Entwicklung des Rollenmodells durch Rahmenbedingungen festgelegt, welche die wichtigsten Eckpunkte und damit auch die Kriterien bei der Entwicklung des Modells darstellen. Das zu entwickelnde Rollenmodell deckt folgende Kriterien ab:

- Unabhängig von einer konkreten Implementierung.
- Unterstützung optimierter Sichten je nach Rolle des Benutzers.
- Bildung der einzigen Instanz für die Autorisierung aller Zugriffe eines implementierenden Systems.
- Möglichkeit des Zugriffs auf unterschiedliche Datenquellen.
- Unabhängigkeit von der Art der Authentifizierung.
- Web-basierte Systeme unterstützen.
- Verteilte Administration ermöglichen.
- Die sehr hohen Sicherheitsanforderungen des Fachbereichs Kerntechnik abdecken.
- Die Einsatzszenarien aus Kapitel 3.1 abdecken.

Diejenigen Anforderungen, die im RBAC-Referenzmodell keine Berücksichtigung finden, werden um ein eigenes Modell ergänzt. Dabei bestand die größte Herausforderung darin, die Rollen nicht nur als reinen Zugriffssteuerungsmechanismus einzusetzen, sondern auch die Sichten eines Benutzers, der eine oder mehrere Rollen im System zugewiesen bekommen hat, direkt beeinflussen zu können.

Während des gesamten Entwicklungsprozesses des in dieser Arbeit vorgestellten Rollenmodells wird darauf geachtet, dass das standardisierte RBAC-Referenzmodell eingehalten wird und die Basis für die Erweiterungen bildet.

3.3 Überführung des RBAC-Modells in ein RBVC-Modell

3.3.1 Begriffsdefinitionen

Die folgenden Begriffsdefinitionen ergänzen die Begriffsdefinitionen des RBAC-Referenzmodells aus Kapitel 2.2.1 und führen die neuen, erweiterten Elemente des Role-Based View Control (RBVC)-Modells ein:

- **Sichten (VIEWS):** Eine Sicht ist die Implementierung des Verhaltens einer bestimmten Ausprägung einer Funktion, die dadurch für eine Rolle optimiert wurde. Der Begriff Sicht bezieht sich auf die gesamte Benutzerschnittstelle. Dazu gehören sowohl das Eingabe- als auch das Ausgabeverhalten und die Visualisierungsmethoden des Systems.
- **Funktionen (FUNCTIONS):** Funktionen repräsentieren im Kontext dieser Arbeit die Gesamtheit der Eigenschaften eines Softwaresystems. Im Gegensatz dazu sind Operationen die ausführbaren Methoden auf Berechtigungsebene (vgl. 2.2.1). Funktionen eines Systems werden für den Benutzer meist durch Menüpunkte dargestellt aber niemals direkt ausgeführt, sondern nur deren rollenabhängige Verhaltensimplementierungen.
- **Einheiten (UNITS):** Zur verteilten Administration werden Einheiten gebildet, die jeweils einem getrennten Verwaltungsbereich entsprechen und hierarchisch gegliedert sein können.

3.3.2 Ressourcentyp

Beim RBAC-Referenzmodell wird implizit angenommen, dass jede Operation auf jedes Objekt ausgeführt werden kann. Auch andere Veröffentlichungen [25]-[27] [34]-[37] folgen dieser Annahme.

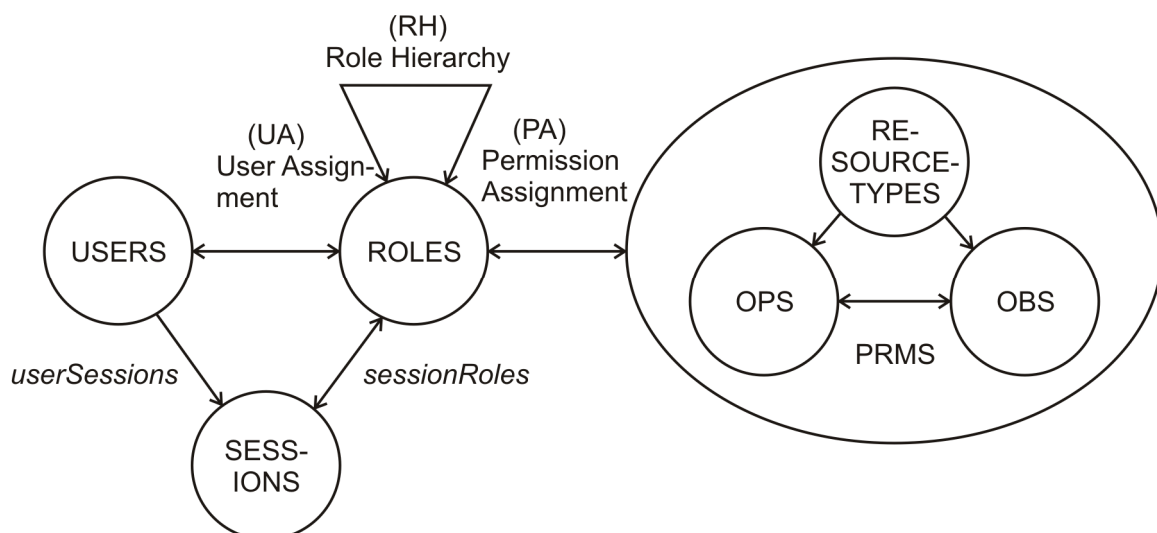


Abb. 12: Erweiterung des RBAC-Modells um Ressourcentypen

Im Gegensatz dazu gibt es im untersuchten Fachbereich Kerntechnik bei komplexen Softwaresystemen sehr spezifische Quellen, aus denen Informationen bezogen werden können. Die Operationen lassen sich daher immer nur auf gleiche Objekte

innerhalb einer Quelle anwenden. Um immer sicherstellen zu können, dass eine Operation auch auf ein Objekt angewandt werden kann, wurden hierzu Ressourcentypen (RESOURCETYPES) eingeführt (Abb. 12).

Ressourcentypen legen die Art der Datenquelle fest. Diese kann beispielsweise

- eine Datei,
- ein Web-Service²³,
- eine Datenbank,
- ein Simulationsprogramm,
- ein Lernobjekt einer Web-based Training (WBT) Anwendung

oder aber auch eine beliebige Kombination aus gleichen oder unterschiedlichen Datenquellen sein.

Als neue wichtige Eigenschaft ermöglichen Ressourcentypen eine Differenzierung der Granularität für die Zugriffe auf Objekte, die für bestimmte, untergeordnete Typen, beispielsweise statische Inhalte, über Platzhalter (Wildcards) zusammengefasst werden können.

Die Ressourcentypen haben jeweils eine Relation zu den Objekten und den Operationen mit der Kardinalität 1:n und bilden somit indirekt auch eine m:n-Beziehung zwischen den Objekten und Operationen ab.

Objekte und Operationen, die einem bestimmten Ressourcentypen angehören, werden abgebildet durch:

$$Obs(rt : RESOURCETYPES) \rightarrow \{Ob : OBS\}, Ops(rt : RESOURCETYPES) \rightarrow \{Op : OPS\}$$

Wenn Objekte und Operationen auf die Ressourcentypen abgebildet werden,

$$resType(ob : OBS) \rightarrow RESOURCETYPES, resType(op : OPS) \rightarrow RESOURCETYPES$$

dann gelten formal für die Zuweisungen zwischen Objekten und Operation die durch die Ressourcentypen eingeschränkten Zusammenhänge:

$$assignableOperations(ob) = \{ob \in OBS, op \in OPS \mid resType(ob) = resType(op)\}.$$

²³ vgl. Kapitel 4.1

In den folgenden Abbildungen des erweiterten Modells wird auf die graphische Darstellung der Ressourcentypen zur besseren Übersichtlichkeit verzichtet. Es wird aber davon ausgegangen, dass diese zur Verfügung stehen.

3.3.3 Rollentyp

Die Einführung von Rollentypen (ROLETYPES) in Abb. 13 erweitert das RBAC-Modell um die Möglichkeit, Einfluss auf das Verhalten von Rollen zu nehmen.

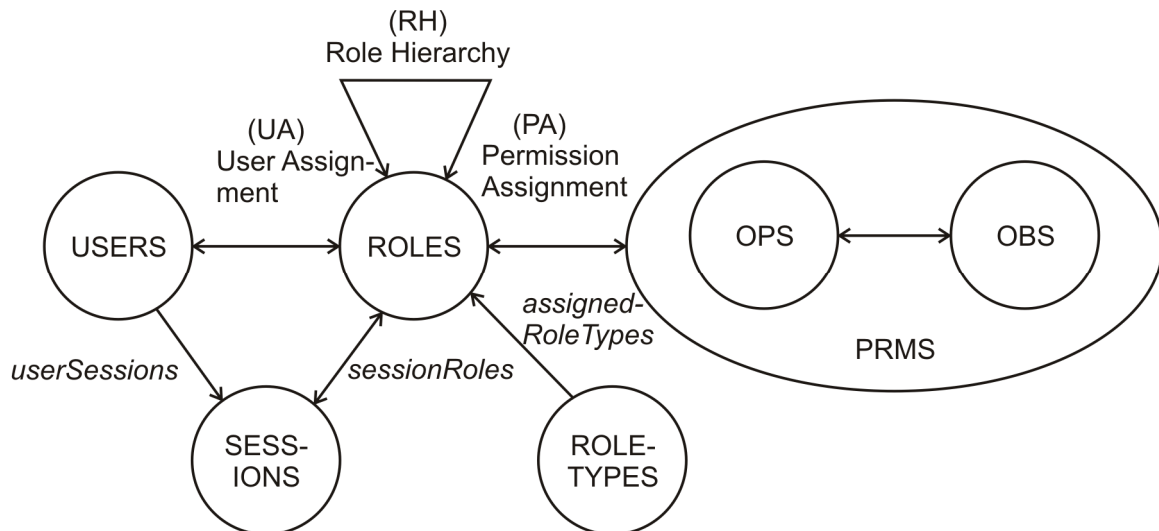


Abb. 13: Erweiterung des RBAC-Modells um Rollentypen

Die ANSI-Norm unterscheidet bei RBAC zweierlei Arten von Hierarchien: *General Hierarchy* und *Limited Hierarchy*. In der Praxis reicht die exklusive Verwendung einer dieser beiden Arten häufig nicht aus. Durch die Einführung der Rollentypen ist es nun möglich, gleichzeitig mehrere Hierarchiearten für unterschiedliche Rollentypen zu unterstützen. Diese neuen Hierarchiearten sind in der Tab. 1 aufgeführt und in Abb. 14 als Hasse-Diagramm abgebildet. Die Namen der Vorgänger beginnen hierbei mit dem Anfangsbuchstaben A, die der Nachfolger mit dem Anfangsbuchstaben D. Eine Vererbungsbeziehung zwischen unterschiedlichen Rollentypen ist nicht möglich.

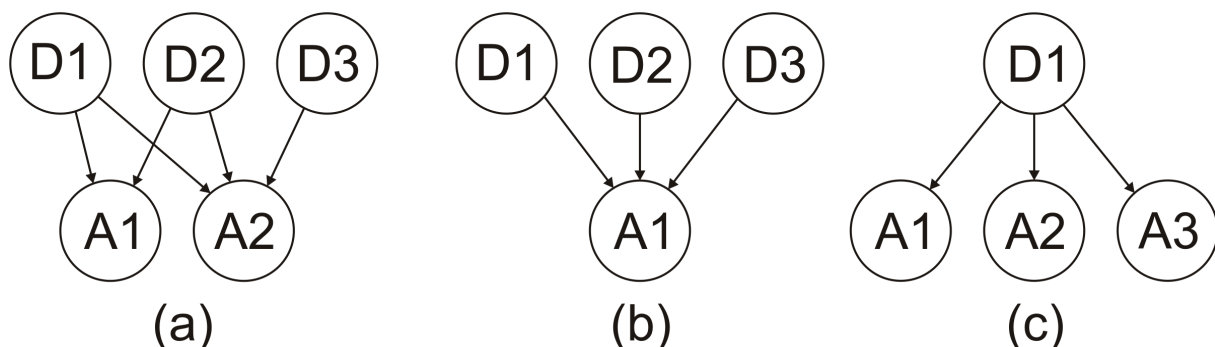


Abb. 14: Erweiterung der Hierarchie-Arten mit Hilfe der Rollentypen

Hierarchie-Art	Beschreibung
General (Abb. 14-a)	Mehrfachvererbung in beide Richtungen
Limited_one_common_ancestor (Abb. 14-b)	Vererbung mit einem gemeinsamen Vorgänger und mehreren Nachfolgern
Limited_one_common_descendant (Abb. 14-c)	Vererbung mit einem gemeinsamen Nachfolger und mehreren Vorgängern
None	Keine Vererbung möglich

Tab. 1: Hierarchie-Arten

Eine weitere Einflussnahme besteht darin, dass man die Vererbung nicht auf die Berechtigungen beschränkt, sondern auch auf Benutzer erweitern kann. Dadurch können Gruppenzugehörigkeiten in diesem neuen Hybrid-RBAC-Modell abgebildet werden (vgl. Kap. 2.1.2), allerdings mit dem entscheidenden Vorteil, dass die Berechtigungen nach wie vor exklusiv den Gruppen-Rollen und niemals den Benutzern zugewiesen werden. Somit lassen sich beispielsweise Benutzerforen in Form von Gruppen aufbauen, die auch hierarchisch gegliedert sein können. Wichtig ist hier die Beschränkung, dass exklusiv entweder Gruppenzugehörigkeiten oder Berechtigungen vererbt werden können.

Die n:1-Abbildung der Rollen auf einen Rollentyp geschieht über

$$assignedRoleType(r : ROLES) \rightarrow ROLETYPES .$$

Zur besseren Übersichtlichkeit wird in den folgenden Abbildungen des erweiterten Modells auf die graphische Darstellung der Rollentypen verzichtet. Es wird aber ebenfalls davon ausgegangen, dass diese zur Verfügung stehen.

3.4 Rollenbasierte Sichtensteuerung

3.4.1 Was ist rollenbasierte Sichtensteuerung?

Rollenbasierte Sichtensteuerung (RBVC) übernimmt neben der Zugriffssteuerung auch die Steuerung der Sichten in einem System. Der Begriff wurde angelehnt an RBAC, da RBVC auf diesem Modell aufbaut.

Der Ansatz zur Einführung des RBVC-Modells (Abb. 15) resultiert aus der Annahme, wenn einem Benutzer eine bestimmte Aufgabe in einer Organisation zugewiesen wird, dann sollte er nicht nur die dafür notwendigen Zugriffsberechtigungen durch Zuweisung einer entsprechenden Rolle erhalten, sondern auch auf seine Aufgabe optimierte Sichten (VIEWS) auf einzelne Funktionsausprägungen des Gesamtsystems.

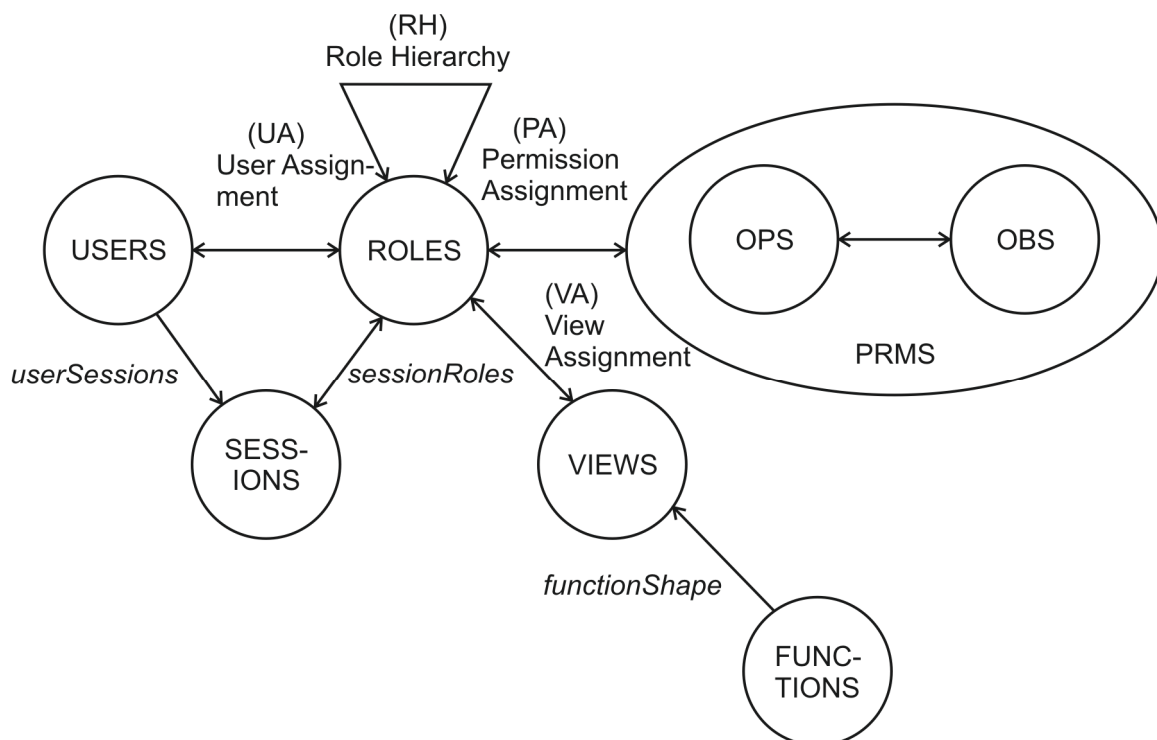


Abb. 15: RBVC-Modell als Erweiterung des RBAC-Modells

Diese Sichten implementieren das Verhalten der Funktionen des Systems, deren Granularität aus den Aufgaben typischer Benutzer in deren Rollen resultieren. Somit ist es möglich, die gleiche Funktion in einer rollenoptimierten Ausprägung anzubieten. Jede Sicht führt durch ihre Implementierung beliebig viele Operationen auf Objekte aus, allerdings wird jede Operation auf ein Objekt vor der Ausführung über die momentan aktive Rolle autorisiert. Deswegen gibt es zwischen Sichten und Operationen in Abb. 15 keine direkte Verbindung.

Die Beziehung VA (View Assignment) mit der Kardinalität m:n bildet die Zugehörigkeit der Sichten (VIEWS) zu den Rollen (ROLES) ab. VA ist eine echte Teilmenge der Produktmenge aus den Sichten mit den Rollen. Sie ist spezifiziert als

$$VA \subseteq VIEWS \times ROLES .$$

Für die Abbildung der Rollen (ROLES) auf eine Menge von Sichten (VIEWS)

$$assignedViews(r : ROLES) \rightarrow 2^{VIEWS}$$

gilt formal:

$$assignedViews(r) = \{v \in VIEWS \mid (v, r) \in VA\}.$$

Es wird definiert, dass FUNCTIONS die Menge an Funktionen eines Gesamtsystems darstellt. Im RBVC-Modell ergibt sich für die Verhaltensimplementierungen der Funktionen in unterschiedlichen Ausprägungen durch Sichten (VIEWS) die Beziehung

$$functionShape(v : VIEWS) \rightarrow FUNCTIONS$$

mit der Kardinalität n:1.

3.4.2 Funktionsauswahl durch den Benutzer

Die Funktionen einer Anwendung werden im Normalfall in Form von Menüs²⁴ den Benutzern angeboten. Im Bereich der Web-Anwendungen sind die Menüs meistens am linken Rand der Anwendung angeordnet.

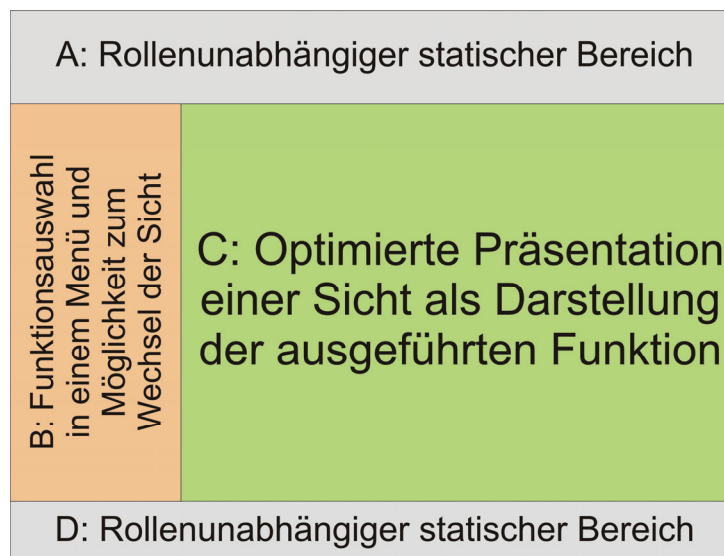


Abb. 16: Exemplarischer Aufbau einer Oberfläche für rollenoptimierte Sichten

Abb. 16 zeigt den exemplarischen Aufbau der Oberfläche einer Web-Anwendung. Die Bereiche A und D sind dabei rollen-unabhängige, statische Bereiche. Sie enthal-

²⁴ Der Begriff Menü stammt vom englischen Begriff *menu* ab und enthält eine Liste von Funktionen, welche vom Benutzer aufgerufen werden können.

ten meist allgemeine Informationen, Logos, etc. Auf der rechten Seite befindet sich der Bereich C, in dem eine optimierte Präsentation einer aktiven Sicht als Ausprägung der aktuell ausgeführten Funktion dargestellt wird. Am linken Rand der Abb. 16 ist der Bereich B angeordnet, der dem Benutzer anhand seiner aktiven Rollen eine Funktionsauswahl in einem Menü ermöglicht. Ein Beispiel eines solchen Menüs ist in Abb. 17 dargestellt. Doch was passiert, wenn einem Benutzer mehrere Rollen zugewiesen werden, denen wiederum Sichten zugewiesen sind, in denen die gleichen Funktionen enthalten sind?

Die Herausforderung besteht in diesem Fall darin, einem Benutzer ein für seine Rollen angepasstes Menü anzubieten, welches ihm einen Zugriff auf alle autorisierten Funktionen ermöglicht. Hat ein Benutzer mehrere Rollen in einem System, die ihm unterschiedliche Ausprägungen der Funktionen in Form von unterschiedlichen Sichten ermöglichen würden, so müssen diese durch eine Kollisionsprüfung festgestellt werden. Das Menü muss nun dem Benutzer die Möglichkeit offerieren, einen Wechsel seiner Rolle für kollidierende Funktionen anzubieten.

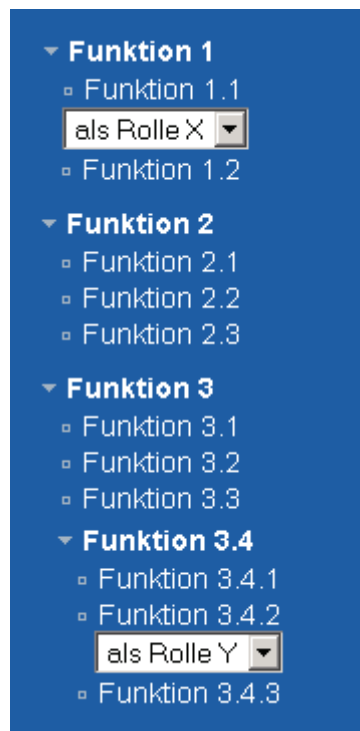


Abb. 17: Beispiel eines Menüs mit kollidierenden Rollen für einzelne Sichten

In Abb. 17 weisen sowohl die Funktion 1.1 als auch die Funktion 3.4.2 Kollisionen auf. Deshalb befinden sich unterhalb dieser Funktionen Auswahllisten, mit deren Hilfe der Benutzer festlegen kann, in welcher Rolle eine Funktion ausgeführt werden

soll. Liegt für eine Funktion keine Kollision vor, so wird sie ohne Auswahlmöglichkeit dargestellt. Die Rolle, in der sie ausgeführt wird, ist für den Benutzer im Normalfall von zweitrangiger Bedeutung und wird deswegen in diesem Beispiel, aber auch in den meisten Anwendungsfällen des RBVC-Modells, nicht dargestellt.

Durch das Auswählen einer Funktion im Menü wird erreicht, dass die dazu notwendige Rolle in der aktuellen Sitzung des Benutzers aktiviert wird. Dabei werden, sofern Interessenkonflikte über DSD-Regeln verhindert werden sollen (vgl. Kap. 2.2.4), diese überprüft und entsprechend angewandt.

3.4.3 Fachkompetenzen und zugewiesene Aufgaben

Wie werden die Rollen festgelegt, die später den Benutzern zugewiesen werden sollen? Welche Sichten werden mit den jeweiligen Rollen verbunden? In welchen Fällen machen unterschiedliche Sichten für verschiedene Rollen einen Sinn? Diese Fragen lassen sich in erster Linie aus den zugewiesenen Aufgaben, welche ein Benutzer in einer Organisation hat, und in zweiter Linie aus seinen Fachkompetenzen ableiten.

Die Fachkompetenz einer Person besteht aus der Fähigkeit, eigenverantwortlich berufstypische Aufgaben zu übernehmen. Sie wird gebildet aus Erfahrung und der Ausbildung der Person. In der Realität unterscheidet sich sehr häufig diese Fachkompetenz von den notwendigen Kompetenzen zur Ausführung einer konkret zugewiesenen Aufgabe. So kann eine Person über weit mehr Fachkompetenz verfügen, als es ihre eigentlichen Aufgaben erfordern. Die Aufgaben, die eine Person in einem Unternehmen oder in einer Organisation übernehmen muss, werden in einer Aufgabenbeschreibung zusammengefasst und geben einen guten Anhaltspunkt für die Einrichtung einer dafür entsprechenden Rolle.

Ein Beispiel hierfür wäre ein System zur Erfassung und Qualitätskontrolle von Fortschrittsberichten verschiedener Forschungsprojekte. Angenommen, eine Person wäre prinzipiell (z.B. durch die Ausbildung, Position im Unternehmen) kompetent, nach einer fachlichen Qualitätsprüfung eine abschließende Berichts freigabe in ihrem Bereich vorzunehmen. Diese Person bekommt in der Realität jedoch meist eine konkrete Aufgabe zugewiesen, beispielsweise die Qualitätsprüfung und abschließende Berichts freigabe für ein konkretes Teilprojekt. Es bietet sich an, diese Aufgabenbeschreibung auf eine Rolle abzubilden. Diese Rolle bekommt die notwendigen Be-

rechtigungen und eine optimierte Sicht zugewiesen, so dass ein typischer Benutzer eben diese Aufgabe erfüllen kann.

Ein typischer Benutzer repräsentiert eine Rolle. Benötigt dieser typische Benutzer in seiner bestimmten Rolle eine andere Sicht auf eine Funktion, als ein anderer in einer anderen Rolle, dann müssen für diese Funktion zwei Sichten mit unterschiedlicher Funktionsausprägung erstellt werden. Gleiches gilt, wenn für eine bestimmte Rolle Ergebnisdaten in einer anderen Form aufbereitet, visualisiert, vereinfacht dargestellt oder irgendwie sonst verändert werden müssen.

Um auf das obige Beispiel zurückzukommen, sollte hier untersucht werden, ob für die abschließenden Berichts freigaben in diesem konkreten Teilprojekt Alleinstellungsmerkmale gelten, die für andere Teilprojekte nicht gelten. Ist dies der Fall, dann wird eine neue Sicht für die Rolle erstellt. Ist dies nicht der Fall, wird die Sicht verwendet, die auch in anderen Teilprojekten Verwendung findet.

Die Praxis zeigt, dass die einer Rolle zugewiesenen Aufgaben, die damit verbundenen Sichten und die damit einhergehenden Berechtigungen über einen längeren Zeitraum als konstant angesehen werden können. Im Gegensatz dazu wechseln Benutzer häufiger die ihnen zugewiesenen Aufgaben. Die Entkopplung der Benutzer von den Berechtigungen und Sichten über die Rollen ermöglicht es, diese Änderungen ohne großen Aufwand umzusetzen.

Ein weiteres Beispiel kommt aus dem Bereich des Notfallschutzes im Rahmen der Kernreaktor-Fernüberwachung. Angenommen, eine Person ist prinzipiell kompetent, im Ereignisfall auf bestimmte Berechnungen zur Simulation der Ausbreitung radioaktiver Spaltprodukte zuzugreifen. Diese Person wird aber in der Realität diese Aufgabe meist nur während bestimmter Bereitschaftszeiten durchführen müssen und somit auch die notwendigen Berechtigungen nur für diesen Zeitraum erhalten.

Durch den Einsatz von RBVC ist die Aufgabenzuweisung durch einfache Rollenzuweisung eines Benutzers möglich, wobei diese auch durch zeit- oder anderweitig gesteuerte Automatismen (z.B. Workflow-Komponenten) verändert werden kann. Bei einem Wegfall oder einer Änderung der zugewiesenen Aufgaben muss nur die Rollenzuweisung angepasst oder aktualisiert werden.

3.5 Verteilte Administration

Im RBAC96-Modell [36] [27] wird zur Administration von RBAC eine eindeutige Trennung zwischen Administratorenrollen und Benutzerrollen vorgenommen. Die Administratorenrollen werden benutzt, um die Benutzerrollen zu verwalten. Das ARBAC97-Modell [38] definiert Bereiche innerhalb einer Rollenhierarchie, in denen administrative Rollen Benutzerrollen verwalten können. Die Zuweisung eines Benutzers zu einer neuen Rolle kann nur unter der Voraussetzung erfolgen, dass er bereits eine Rolle zugewiesen bekommen hat, die sich in der Rollenhierarchie unterhalb der neu zuzuweisenden Rolle befindet. Soll eine Berechtigung zu einer Rolle hinzugefügt werden, so muss diese Berechtigung bereits einer anderen Rolle mit höherer Hierarchiestufe zugewiesen sein. Das ARBAC02-Modell [39] erweitert das ARBAC97-Modell, indem es die Voraussetzungen der Rollenadministration nicht mehr von der Hierarchie, sondern von externen Organisationsstrukturen abhängig macht.

Der Ansatz für die verteilte Administration des RBVC-Modells geht ebenfalls von einer externen Struktur aus, welche die Administrationsbereiche in hierarchische Einheiten gliedert. Dieser Ansatz soll verteilte Administration ermöglichen, die es einzelnen Benutzern mit administrativen Berechtigungen erlaubt, Teilbereiche eines Systems zu verwalten. Die Umsetzung der verteilten Administration im RBVC-Modell basiert daher auf der Annahme, dass sich die Administrationsbereiche aus hierarchischen Einheiten zusammensetzen, in denen eine Verwalterrolle die Verantwortlichkeit übernimmt. Durch die Einführung der Rollentypen (vgl. 3.3.3) ist es möglich, die Rollen für unterschiedliche Hierarchiearten parallel zu verwenden. Diese Möglichkeit bildet die Basis, das RBAC-Referenzmodell um administrative Einheiten (UNITS) zu erweitern (Abb. 18). Die Einheiten können sich in einer Hierarchie mit der Kardinalität 1:n befinden und trennen so die Verantwortlichkeiten in Unterbereiche auf, wie sie auch in normalen Projektstrukturen mit mehreren Projektpartnern zu finden sind. Jede Unit hat eine m:n-Beziehung zu den Rollen, über welche die Benutzerrollen für eine Einheit definiert und angelegt werden können. Da die Einheiten bereits über eine eigene Hierarchie verfügen, bekommen die entsprechenden Benutzerrollen einen Rollentyp zugewiesen, der keine Rollenhierarchie (Hierarchie-Art None, vgl. 3.3.3) zulässt.

Jede Einheit ist ein Objekt des RBVC-Modells. Die administrativen Berechtigungen für jede Einheit werden einer separaten Verwalterrolle zugewiesen, die als hierarchi-

sche Vorgängerrolle zu der Rolle angelegt wird, welche die Einheit erstellt hat. Der Rollentyp der Verwalterrollen hat die Hierarchie-Art One_common_descendant. Die Verwalterrolle des Erstellers einer neuen Einheit erbt durch dieses Vorgehen alle Berechtigungen, die auch die Verwalterrolle der neuen Einheit hat.

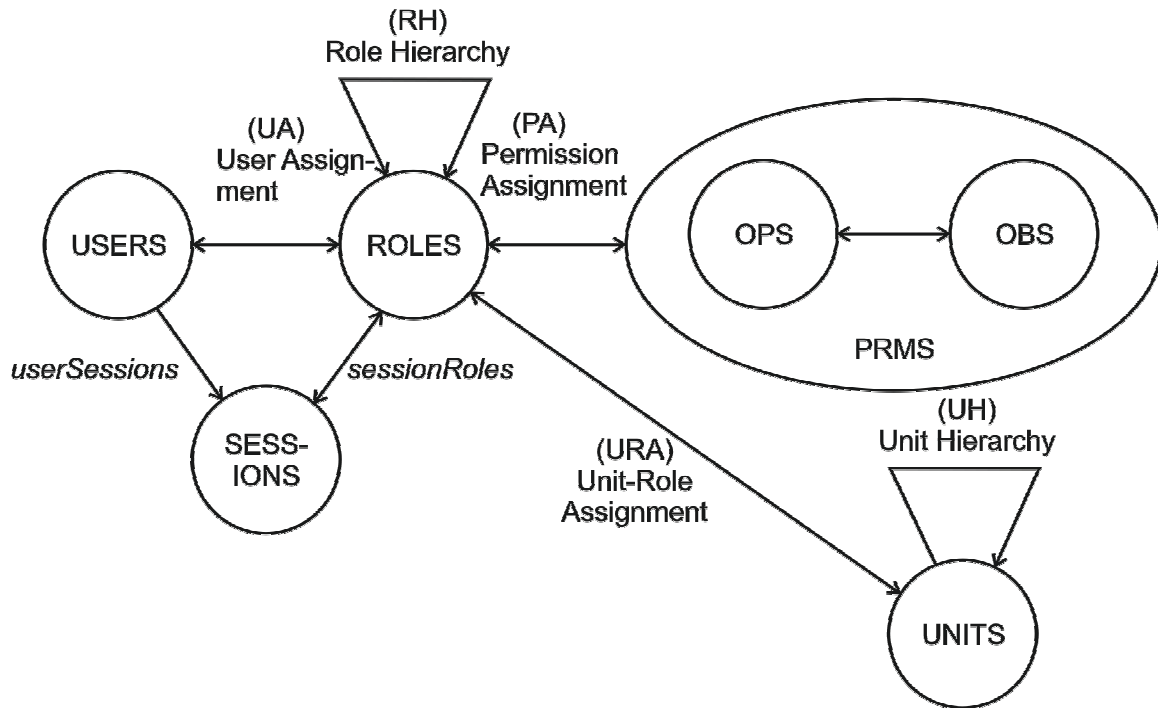


Abb. 18: Einheiten zur Administration von RBVC

Die Beziehung URA (Unit-Role Assignment) mit der Kardinalität m:n bildet die Zugehörigkeit der Rollen (ROLES) zu den Einheiten (UNITS) ab. URA ist eine echte Teilmenge der Produktmenge aus den Rollen mit den Einheiten. Sie ist spezifiziert als

$$URA \subseteq ROLES \times UNITS.$$

Für die Abbildung der Einheiten (UNITS) auf eine Menge von Rollen (ROLES)

$$assignedRoles(un : UNITS) \rightarrow 2^{ROLES}$$

gilt formal:

$$assignedRoles(un) = \{r \in ROLES \mid (r, un) \in URA\}.$$

Die 1:n-Beziehung UH (Unit Hierarchy) aus Abb. 18 ist eine Halbordnung der Produktmenge der Einheiten. Sie ist spezifiziert als

$$UH \subseteq UNITS \times UNITS$$

mit der Hierarchie-Einschränkung auf einen gemeinsamen Nachfolger:

$$\forall u, u_1, u_2 \in UNITS, u_1 \geq u \wedge u_2 \geq u \Rightarrow u_1 = u_2.$$

Der Einsatz der verteilten Administration sollte immer dann vorgesehen werden, wenn verschiedene Benutzer eigenverantwortlich unterschiedliche Einheiten oder Teilbereiche bearbeiten, und diese Untereinheiten einer hierarchischen Struktur sind. Ein Beispiel hierfür wäre ein Forschungsprojekt, bei dem Partner aus verschiedenen Organisationen gemeinsam ein Dokumentverwaltungssystem für die Projektabwicklung nutzen. Solche Projekte werden häufig in Unterprojekte aufgeteilt, in denen die notwendigen Arbeiten in Pakete unterteilt werden. Eine solche Struktur lässt sich auf Einheiten abbilden und kann somit verteilt von den verantwortlichen Projektpartnern verwaltet werden.

3.6 Authentifizierung

Neben der Autorisierung über das vorgestellte Rollenmodell bleibt noch die Frage der Authentisierung²⁵ und der Authentifizierung [40], [41] eines Benutzers offen. Jeder Benutzer muss sich authentifizieren, so dass seine eindeutige Identität festgestellt werden kann, bevor die Methoden für die Autorisierung des Rollenmodells Anwendung finden können. In der Literatur werden drei unterschiedliche Arten der Authentifizierung unterschieden: Wissen, Besitz, biometrisches Merkmal. Um die Sicherheit der Authentifizierbarkeit zu erhöhen werden diese drei Arten sehr häufig auch in beliebigen Kombinationen verwendet.

3.6.1 Authentifizierung durch Wissen

Die Authentifizierung durch Wissen ist eine der gängigsten Methoden, eine Identität festzustellen. Tab. 2 zeigt die Vor- und Nachteile dieser Methode. Nachfolgend werden Beispiele dieser Art der Authentifizierung gegeben.

²⁵ Im Englischen werden die beiden deutschen Begriffe *Authentisierung* und *Authentifizierung* durch denselben Begriff *authentication* übersetzt. Die beiden Begriffe unterscheiden sich dahingehend, dass die *Authentisierung* den Nachweis der eigenen Identität beschreibt, während die *Authentifizierung* die Überprüfung der Identität einer Person oder eines Computersystems beschreibt. In der Literatur werden die beiden Begriffe synonym verwendet, jedoch überwiegt der Begriff *Authentifizierung*.

Beispiele von Methoden zur Authentifizierung durch Wissen

- **Password:** Die Kombination aus Benutzernamen und Passwort ist die am häufigsten eingesetzte Methode zur Authentifizierung eines Benutzers. Leider ist dieser Schutz nur dann als ausreichend anzusehen, wenn der Benutzer das Passwort geheim hält und durch die eingesetzte Software dazu angehalten wird, keine *schwachen Passwörter*²⁶ zu verwenden.
- **Personal Identification Number (PIN):** PIN-Nummern werden häufig im Zusammenhang mit Maestro-Karten²⁷ am Bankautomaten oder zur Anmeldung am Mobiltelefon verwendet, um dadurch die Identitätsfeststellung vorzunehmen. Da PIN-Nummern in der Regel aus relativ kurzen Zahlen bestehen, müssen Methoden vorgesehen werden, dass eine Nummer nicht erraten werden kann. Deshalb sperren Systeme, die eine Authentifizierung über PIN-Nummern machen, nach einer kleinen Anzahl von Fehlversuchen den Zugang.

Vorteile	Nachteile
<ul style="list-style-type: none">• einfache Umsetzbarkeit• kann meist kostengünstig erneut erzeugt werden	<ul style="list-style-type: none">• kann vergessen werden• kann von Dritten abgehört werden• kann an Dritte weitergegeben werden

Tab. 2: Vor- und Nachteile der Authentifizierung durch Wissen

3.6.2 Authentifizierung durch Besitz

Tab. 3 zeigt die Vor- und Nachteile der gebräuchlichsten Authentifizierungsmethoden durch Besitz. Nachfolgend werden Beispiele dieser Art der Authentifizierung gegeben.

²⁶ Schwache Passwörter sind solche, die durch einen Wörterbuchangriff auf einfache Weise erraten werden können [42].

²⁷ Maestro-Karten sind der Nachfolger von EC-Karten

Beispiele von Methoden zur Authentifizierung durch Besitz

- Schlüssel
- Kryptographischer Schlüssel
- Karte mit Magnetstreifen oder Chip
- Radio Frequency Identification (RFID)

Vorteile	Nachteile
<ul style="list-style-type: none">• Vervielfältigung schwierig• kann zeitweise einer anderen Person übergeben werden²⁸	<ul style="list-style-type: none">• kann verloren gehen oder gestohlen werden• benötigt spezielle, meist kostenintensive Fertigungsprozesse• kann zeitweise einer anderen Person übergeben werden

Tab. 3: Vor- und Nachteile der Authentifizierung durch Besitz

3.6.3 Authentifizierung durch biometrisches Merkmal

In Tab. 4 sind die Vor- und Nachteile der gebräuchlichsten Authentifizierungsmethoden durch biometrische Merkmale [43] [44] dargestellt. In den letzten Jahren wurde diese Methode der Identitätsfeststellung immer häufiger angewandt. Beispielsweise findet man heute in tragbaren Computern häufig einen eingebauten Sensor für die biometrische Erkennung. Nachfolgend werden Beispiele dieser Art der Authentifizierung anhand einiger Beispiele gegeben.

Beispiele von Methoden zur Authentifizierung durch biometrisches Merkmal

- Fingerabdruck
- Gesichtsmerkmalerkennung
- Netzhaut- oder Irisscan

²⁸ Dies widerspricht zwar dem zugrundeliegenden Rollenmodell, aber eine solche Möglichkeit wird von Anwendern immer wieder gewünscht. Dieser Punkt sollte daher im Normalfall als Nachteil gewertet werden.

Vorteile	Nachteile
<ul style="list-style-type: none"> • kann nicht vergessen werden • kann in der Regel nicht verloren werden 	<ul style="list-style-type: none"> • benötigt kostenintensive Scanner- bzw. Erkennungstechnik • teilweise Lebenderkennung notwendig • falsche Erkennungen möglich • falsche Zurückweisung möglich

Tab. 4: Vor- und Nachteile der Authentifizierung durch biometrische Merkmale

3.6.4 Authentifizierung in Bezug auf das RBVC-Modell

Die Authentifizierung dient lediglich dazu, die eindeutige Identität eines Benutzers festzustellen. Nachdem die Identität eines Benutzers über ein vorher festgelegtes Verfahren festgestellt wurde, ist er unter einer eindeutigen Identität am System authentifiziert. Schafft es ein potentieller Angreifer, einen Authentifizierungsmechanismus zu umgehen, bekommt er automatisch dieselben Rechte im System, die auch der eigentliche Inhaber der Identität im System hat. Deshalb ist eine sichere Authentifizierung für sensible, z.B. kerntechnische Systeme, von erhöhter Wichtigkeit und sollte in diesem Fall nicht nur von einem Kriterium abhängig gemacht werden. Wünschenswert wäre mindestens eine 2-Faktor-Authentifizierung [45], also eine Kombination aus zwei unterschiedlichen Authentifizierungsmethoden. Für sicherheitskritische Systeme (z.B. Notfallschutz) wäre beispielsweise eine Kombination aus Benutzername und Passwort ergänzt durch einen kryptographischen Schlüssel denkbar und sinnvoll.

Bei der Entwicklung des RBVC-Modells wurde von einer Kombination aus Benutzername und Passwort als Authentifizierungsmechanismus ausgegangen, jedoch wurde darauf geachtet, dass prinzipiell auch jede andere Art der Authentifizierung sowie eine Kombination unterschiedlicher Verfahren jederzeit eingesetzt werden kann, sofern daraus eine eindeutige Identität des aktuellen Benutzers hervorgeht. Dabei ist es nicht entscheidend, ob es sich bei dem Benutzer um einen Menschen oder eine Maschine (z.B. Web-Service), handelt, sofern die festgestellte Identität des Benutzers (USER) im System existiert.

3.7 RBVC-Gesamtmodell

In Abb. 19 sind zusammenfassend sämtliche Erweiterungen dargestellt, welche im Rahmen dieser Arbeit bei der Überführung des RBAC-Referenzmodells in das RBVC-Gesamtmodell eingeführt wurden. Die Elemente des RBAC-Referenzmodells sind in schwarz, die Erweiterungen rot dargestellt. Die Erweiterung Ressourcentypen (RESOURCETYPES) erlaubt die Steuerung des Verhaltens dahingehend, dass Operationen immer nur auf gleiche Objekte innerhalb einer spezifischen Quelle angewandt werden können.

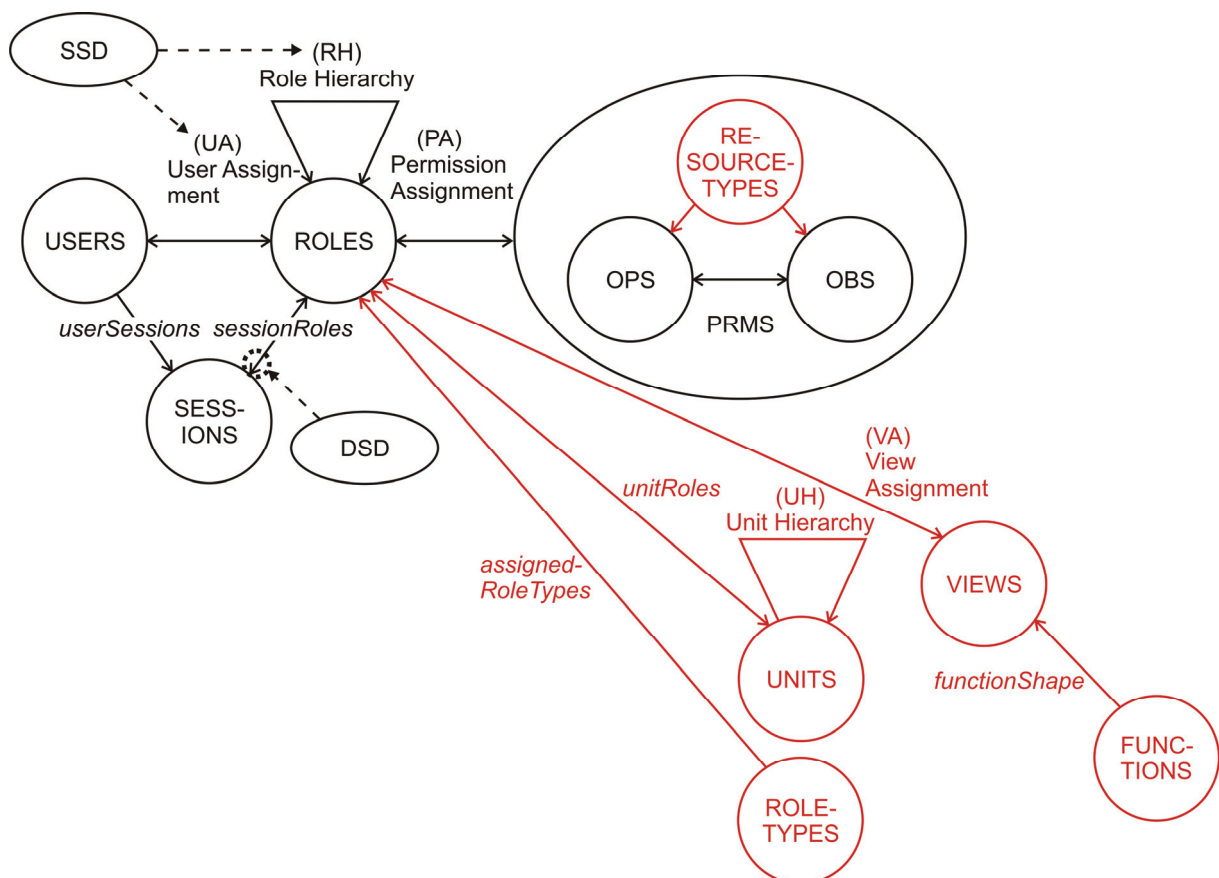


Abb. 19: RBAC-Referenzmodell (schwarz) und Erweiterungen (rot) zur Überführung in das RBVC-Gesamtmodell

Die anderen Erweiterungen sind direkt mit den Rollen verbunden und verdeutlichen deren zentrale Position im RBVC-Gesamtmodell. Neben der Entkopplung der Berechtigungen (PRMS) von den Benutzern (USERS), welche bereits im Referenzmodell vorgesehen ist, übernehmen sie im erweiterten Modell auch die Steuerung des rollenabhängigen Verhaltens der Funktionen (FUNCTIONS) des Systems, welche durch optimiert ausgeprägte Sichten (VIEWS) den Benutzern zur Verfügung gestellt werden.

Durch die Einführung der Rollentypen (ROLETYPES) kann Einfluss auf die beschriebenen Hierarchiearten genommen werden. Im Gegensatz zum RBAC-Referenzmodell ist im RBVC-Modell die gleichzeitige Verwendung mehrerer dieser Hierarchiearten vorgesehen. Aufbauend auf diese neue Möglichkeit wurden Einheiten (UNITS) eingeführt, welche eine verteilte Administration in hierarchisch gegliederten Teilbereichen erlauben.

4 Konzept und Implementierung einer Basisarchitektur für Web-Anwendungen unter Berücksichtigung des RBVC-Modells

4.1 Definition der Funktionsanforderungen

Im ersten Schritt wurde eine Analyse bestehender Softwaresysteme²⁹ für den Fachbereich Kerntechnik durchgeführt, um einen Überblick über die bereits bestehenden Funktionen dieser Systeme zu erhalten. Dabei wurden auch Verbesserungswünsche, die im Laufe des Betriebs der bestehenden Systeme entstanden sind, eingearbeitet.

Darauf aufbauend wurden die Einsatzszenarien des Rollenmodells in der Kerntechnik aus Kapitel 3.1 analysiert um die Funktionen und Architekturmerkmale der Basisarchitektur definieren zu können. Daraus resultierte eine Liste mit Rahmenbedingungen, die sich wie folgt zusammenfassen lässt. Die Basisarchitektur soll:

- Über Komponenten erweitert werden können.
- Das RBVC-Modell für sämtliche Autorisierungen verwenden.
- Das RBVC-Modell für die Generierung rollenoptimierter Sichten verwenden.
- Eine Schnittstelle für die Authentifizierung anbieten.
- Unterschiedliche Datenquellen unterstützen.
- Entkoppelt sein von Datenbank-Dialekten.
- Als Komponente eingebundene Web-Anwendungen zur Verfügung stellen können:
 - Eine Schnittstelle besitzen, über die man bestehende Simulationsprogramme, die auch in anderen Programmiersprachen erstellt wurden, einbinden kann.
 - Eine Schnittstelle besitzen, über die man Lehr-/Lernsysteme anbinden kann.
 - Eine Schnittstelle besitzen, über die man Wissens- und Dokumentmanagementfunktionen anbinden kann.

²⁹ Diese Softwaresysteme (vgl. Kap. 1.6) wurden in den vergangenen Jahren am Institut für Kerntechnik und Energiesysteme entwickelt und werden heute in Produktionsumgebungen eingesetzt.

- Eine Schnittstelle für die Anbindung von Workflow-Engines enthalten.
- Web-Service-Schnittstellen³⁰ [46] [47] unterstützen.
- Einfache Konsistenzprüfungen bei Formulareingaben durchführen können.

4.2 Mehrschichtarchitektur

Für die Umsetzung der Basisarchitektur wurde eine Dreischichtarchitektur [48]-[51] gewählt (Abb. 20), die aus den drei Schichten

- Präsentationsschicht (Presentation Tier)
- Logikschicht³¹ (Logic Tier)
- Datenschicht (Data Tier)

besteht. Der Vorteil einer Dreischichtarchitektur im Vergleich zu einer Architektur, die auf dem *Model View Controller* (MVC)-Ansatz beruht, liegt darin, dass die Präsentationsschicht streng von der Datenschicht getrennt wird. Die gesamte Kommunikation zwischen den beiden Schichten wird durch die Logikschicht entkoppelt. Beim MVC-Ansatz hingegen wird dagegen die Präsentation, welche der Präsentationsschicht entspricht, vom Modell, welches der Datenschicht entspricht, direkt aktualisiert.

Um das Systemverhalten zu beschreiben wird ein Überblick über den Ablauf am Beispiel einer einzelnen Benutzerinteraktion gegeben. Es wird davon ausgegangen, dass die Authentifizierung des Benutzers bereits stattgefunden hat, vom Sichtengenerator die Start-Ansicht erstellt wurde und diese in der Präsentationsschicht dargestellt ist.

Die Präsentationsschicht besteht aus einem Browser, der über das Internet an die Logikschicht angebunden wird. Der Benutzer wählt eine Funktion aus worauf der Browser eine Anfrage an die Logikschicht sendet, die diese nach erfolgter Autorisierung durch die Zugriffssteuerung des RBVC-Modells in der Steuerung der Logikschicht verarbeitet und an die jeweils zuständige Anwendung weiterleitet. Die Anwendung führt die notwendigen Interaktionen mit der Datenschicht durch und übergibt das Ergebnis der Anfrage an die Sichtensteuerung, die mit Hilfe des Sichtengenerators die Antwort generiert und an den Browser zurücksendet.

³⁰ Ein Web-Service dient der direkten Kommunikation zwischen Anwendungen über das Internet

³¹ Die Logikschicht wird in der Literatur auch häufig mit dem Begriff *Business Tier* bezeichnet

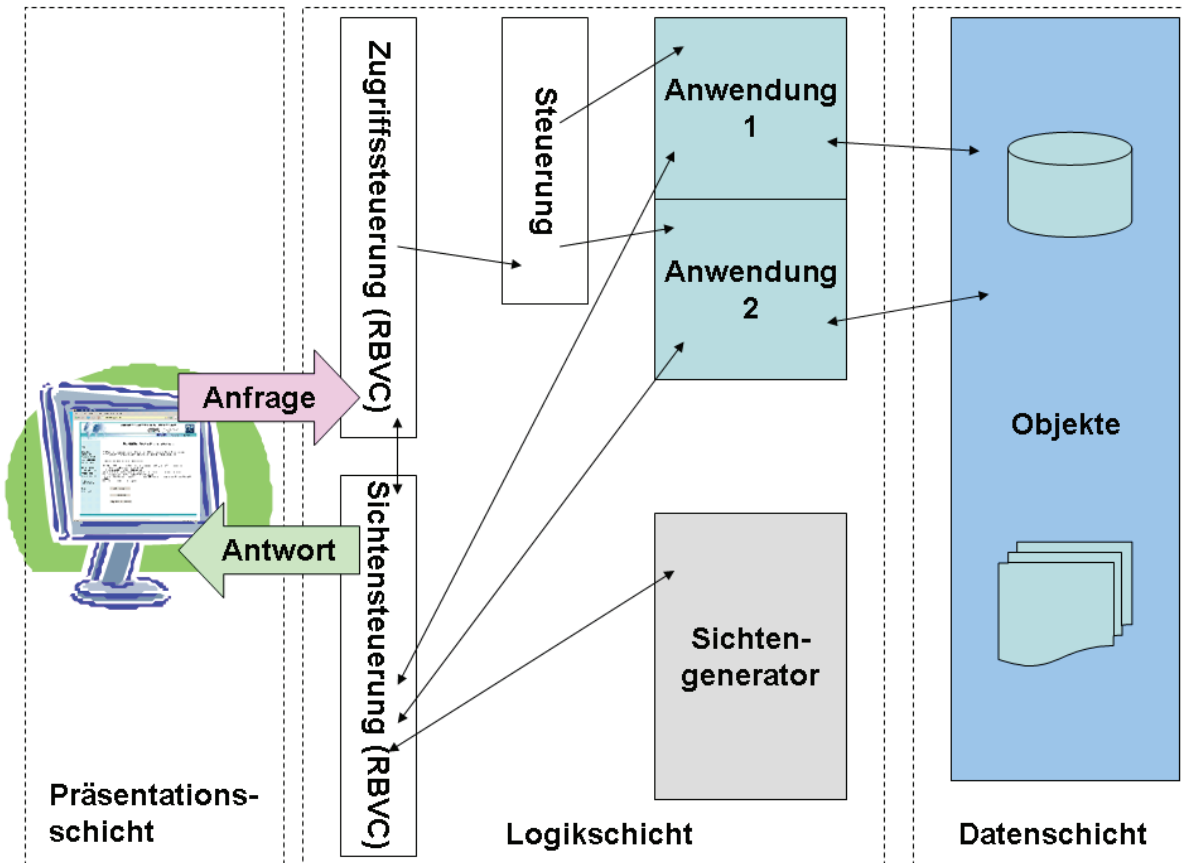


Abb. 20: Dreischichtmodell der Basisarchitektur mit Unterstützung von RBVC
(konventionelle Darstellung)

Neben der konventionellen Darstellung der Systemarchitektur in Abb. 20 wird in Abb. 21 die direkte Abbildung des RBVC-Modells auf das Dreischichtmodell der Basisarchitektur gezeigt.

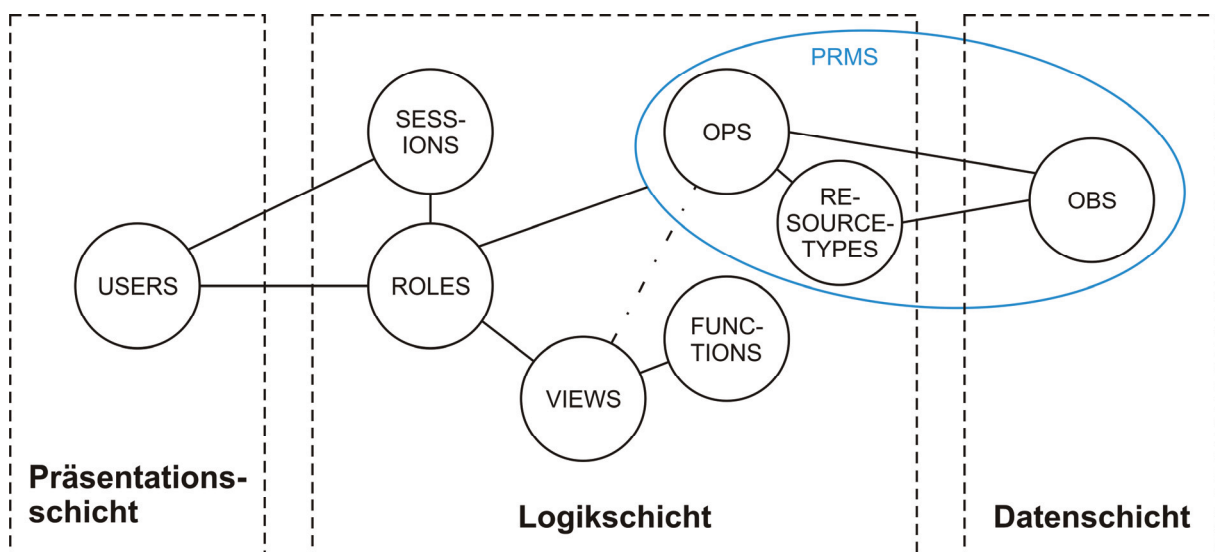


Abb. 21: Dreischichtmodell der Basisarchitektur mit Unterstützung von RBVC
(direkte Abbildung des Rollenmodells)

Der Hauptteil des RBVC-Modells wird durch die Logikschicht umgesetzt, lediglich die Benutzer (USERS) werden durch die Präsentationsschicht und die Objekte (OBS), welche Bestandteil der Berechtigungen (PRMS) sind, durch die Datenschicht repräsentiert.

In den folgenden Unterkapiteln wird auf die Funktionsweise der einzelnen Schichten näher eingegangen.

4.2.1 Präsentationsschicht

Die Präsentationsschicht stellt die Benutzerschnittstelle der Web-Anwendung dar, die sowohl für die Aus- aber auch für die Eingaben verantwortlich ist. Sie wird durch einen Browser [52] [53] repräsentiert.

Im Browser können gewisse einfache Überprüfungen (z.B. Typprüfungen für Formularfelder) eigenständig durchgeführt, ohne dass dafür eine Server-Interaktion über das Internet notwendig wäre. Optional ist vorgesehen, dass AJAX (Asynchronous JavaScript and XML) [54] für die Kommunikation zwischen Präsentationsschicht und Logikschicht verwendet werden kann.

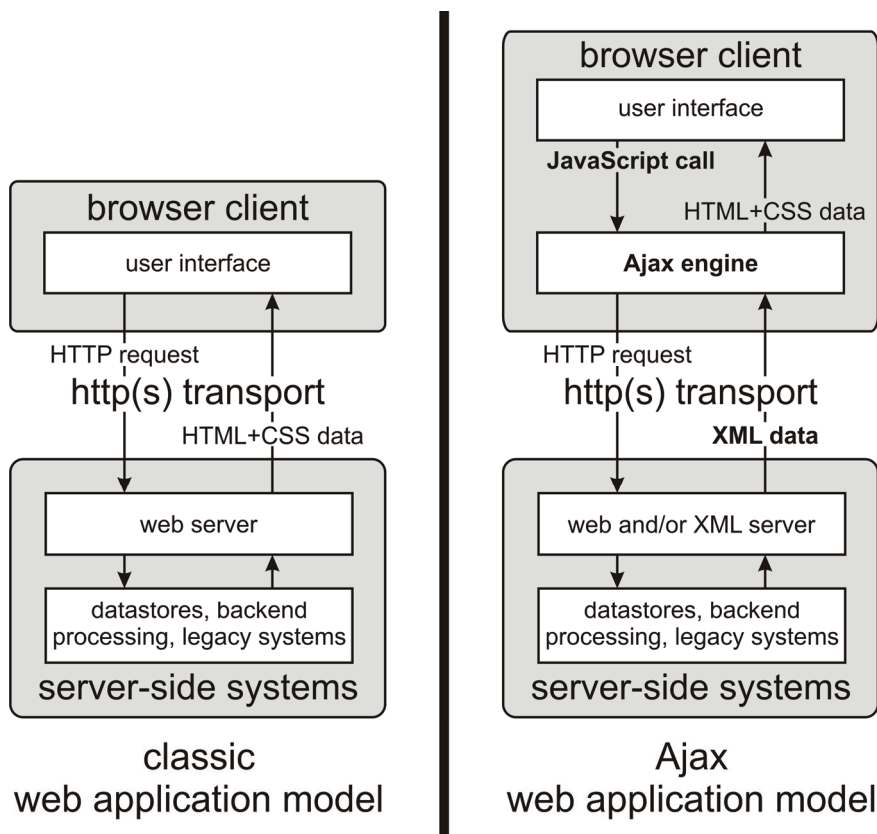


Abb. 22: Vergleich der Klient-Server-Interaktion bei klassischer Web-Anwendung und bei AJAX-Web-Anwendung [54]

In Abb. 22 wird die Klient-Server-Interaktion bei einer klassischen Web-Anwendung der Interaktion bei einer AJAX-Web-Anwendung gegenübergestellt. Die Kommunikation zwischen Klient und Server erfolgt in beiden Fällen über das *Hypertext Transfer Protocol* (http) [55] oder bevorzugt über das *Hypertext Transfer Protocol Secure* (https) [56], welches die mit *Secure Sockets Layer* (SSL) [57] verschlüsselte Variante des erstgenannten Protokolls ist.

In klassischen Web-Anwendungen stößt jedes klientenseitige Ereignis der Benutzerschnittstelle (user interface) eine Anfrage zum Web-Server an. Dieser bearbeitet die Anfrage, indem er die notwendigen Informationen für eine Antwort bei den nachgestellten Systemen anfragt. Der Web-Server liefert daraufhin das Ergebnis an den Browser in Form eines *Hypertext Markup Language* (HTML)-Dokumentes³² [58] als Antwort zurück.

Im Gegensatz dazu werden bei AJAX-Web-Anwendungen *Extensible Markup Language* (XML)-Daten [59] zur Kommunikation zwischen Klient und Server eingesetzt. Die klientenseitige Umwandlung der XML-Daten in vom Browser darstellbare Daten wird von der JavaScript-basierten [60] [61] AJAX-Engine umgesetzt, welche die XML-Daten in das darstellbare HTML-Format umwandelt. Auf Seiten des Web-Servers müssen daher beim Einsatz von AJAX keine aufwendigen HTML-Dokumente erstellt werden, sondern es können direkt XML-Daten ausgetauscht werden. Die nachgestellten Systeme des Web-Servers unterscheiden sich nicht von denen der klassischen Web-Anwendung.

In Abb. 23 ist der zeitliche Verlauf der synchronen Klient-Server-Interaktion bei einer klassischen Web-Anwendung und der asynchronen Interaktion einer AJAX-Web-Anwendung gegenübergestellt. Bei der klassischen Web-Anwendung erfolgt nach jeder Benutzeraktivität (user activity) ein Transfer (data transmission) zwischen Klient und Server. Während der Zeit, in der der Server die Anfrage bearbeitet (system processing), wartet die klassische Web-Anwendung und kann vom Benutzer nicht verwendet werden. Die AJAX-Web-Anwendung entkoppelt diese Anfragen, indem die Benutzeraktivität (user activity - input) die AJAX-Engine zu einer Server-Anfrage (data transmission) veranlasst, die Ansicht des Benutzers aber klientenseitig aktualisiert

³² Das zurückgelieferte HTML-Dokument kann dabei optional mit einer Formatierungsdatei im Cascading Style Sheets (CSS)-Format [60] kombiniert sein.

(display) werden kann, bevor die Antwort des Servers (server-side processing, data transmission) zurückkommt.

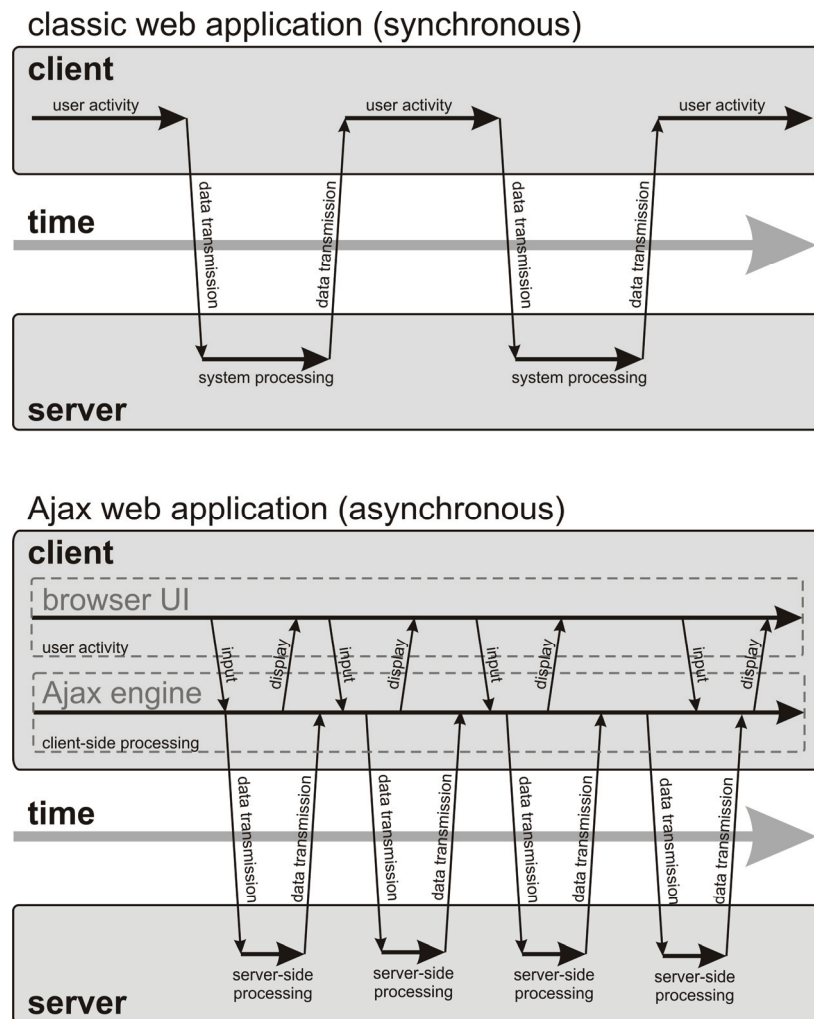


Abb. 23: Vergleich synchroner und asynchroner Kommunikation [54]

Der große Vorteil von AJAX-Web-Anwendungen liegt also darin, dass durch eine Beschränkung auf die Übermittlung der reinen XML-Daten auf der einen Seite deutlich weniger Transfermenge anfällt und die Anwendung durch das asynchrone Verhalten für den Benutzer flüssiger wird. Auf der anderen Seite muss aber ein Teil der Formular-Steuerungsfunktionen in den Browser verlegt werden. Die Verwendung von AJAX verlagert somit letztendlich den Teil der Logikschicht, der die Formulare betrifft, in die Präsentationsschicht.

4.2.2 Logikschicht

Für die folgende Beschreibung der Logikschicht wird davon ausgegangen, dass ein Benutzer bereits am System angemeldet ist. Er wurde bereits authentifiziert und seine Identität im Rahmen einer laufenden Sitzung (SESSIONS) gespeichert. Nachdem

er eine Funktionsauswahl in seinem Menü getroffen hat, wird die erste Seite der dadurch gewählten Sicht (VIEWS) in der Präsentationsschicht dargestellt. Die Sicht implementiert dabei das Verhalten einer Funktion (FUNCTIONS) des Systems in einer rollenoptimierten Ausprägung.

Setzt der Benutzer nun seine Arbeit fort und führt eine Aktion aus, dann empfängt die Logikschicht die Anfrage der Präsentationsschicht und muss diese bearbeiten (Abb. 20 und Abb. 21). Die Zugriffssteuerung des RBVC-Modells nimmt anhand der Identität des Benutzers (USERS) die Autorisierung der Anfrage vor. Dazu wird überprüft, ob der Benutzer die im Rahmen der aktuellen Sicht benötigten Berechtigungen (PRMS) hat. Intern führt die Aktion als Teil der Sicht Operationen (OPS) auf ein oder mehrere Objekte (OBS) aus, diese werden aber in jedem Einzelfall über die Rollen und den damit verbundenen Berechtigungen (PRMS) autorisiert. Die Anwendbarkeit einer Operation auf ein Objekt ist dabei eindeutig durch die Ressourcentypen (RESOURCETYPES) bestimmt.

Hat ein Benutzer aufgrund seiner Rollen die Berechtigung, die angefragte Operation auf das Objekt auszuführen, dann werden die Informationen der Anfrage an die Steuerung (Abb. 20) übergeben, welche die Abläufe der Benutzerschnittstelle steuert, interne Zustände speichert und die Daten an die richtige Anwendung, die als Komponente eingebunden ist, weiterleitet.

Eine Komponente in diesem Kontext beschreibt eine Software, die über eine spezifizierte Schnittstelle, die sogenannte Komponentenschnittstelle, an die Logikschicht angebunden wird. Die Komponentenschnittstelle ist so gestaltet, dass unterschiedliche Arten von Anwendungen über die gleiche Schnittstelle zugreifen können, da diese keine komponentenspezifischen Elemente enthält.

Die Anwendung führt die empfangene Anweisungen aus und greift dabei falls notwendig auf die Datenschicht zu, um dort Daten zu beschaffen oder zu speichern. Danach übergibt die Anwendung das Ergebnis an die Sichtensteuerung des RBVC-Modells, welche vom Sichtengenerator eine weitere Darstellung der Sicht entsprechend der Rolle des Benutzers erstellen lässt und diese als Antwort an die Präsentationsschicht zurückgibt.

4.2.3 Datenschicht

Die Datenschicht kann aus unterschiedlichen Datenquellen bestehen, in denen die Objekte abgelegt werden. Als Datenquelle für Objekte können daher nicht nur Datenbanken und Dateien, sondern auch andere Systeme fungieren.

Für relationale Datenbanken wird ein Persistenz-Framework angebunden, das für die Persistenz der Daten auch bei längeren und konkurrierenden Transaktionen verantwortlich ist. Diese Schicht erzeugt eine Objekt-Relationale-Abbildung³³ [62], wodurch keine direkten Zugriffe mehr auf die Datenbank benötigt werden, sondern direkt mit Objekten gearbeitet werden kann. Weiter ist diese Schicht verantwortlich für die Unabhängigkeit von den diversen Datenbankdialekten.

4.3 Programmiersprache und Entwicklungsumgebung

Die Festlegung auf eine Programmiersprache und eine Entwicklungsumgebung stellte neben der Auswahl von weiterführenden Technologien eine der wichtigsten Grundlagen für die Implementierung der Basisarchitektur dar.

Die Programmiersprache muss rein objektorientiert sein und plattformunabhängig zur Verfügung stehen. Weiter soll sie über eine große Anzahl freier Bibliotheken verfügen, sowie über ein Framework für Web-Anwendungen.

Die Wahl der Programmiersprache fiel auf Java, da diese die vorher beschriebenen Kriterien am besten erfüllt. Die Wahl der Entwicklungsumgebung fiel auf das frei verfügbare Eclipse, da es eine optimale Unterstützung für die Entwicklung mit der Programmiersprache Java anbietet und die Methoden des Refactorings³⁴ sehr gut unterstützt. Die Entwicklungsumgebung hat bereits im Standardumfang die Java-Testumgebung JUnit integriert, die eine testgetriebene Entwicklung ermöglicht.

In den beiden folgenden Unterkapiteln werden Java als Programmiersprache und Eclipse als Entwicklungsplattform einführend vorgestellt.

³³ In der Literatur wird hierfür auch häufig der englische Begriff *Object-Relational-Mapping* verwendet.

³⁴ Mit dem Begriff *Refactoring* [65] wird eine Strukturverbesserung von Quelltexten bezeichnet, ohne dabei eine Änderung des Programmverhaltens hervorzurufen.

4.3.1 Java

Im Jahr 1995 wurde von Sun Microsystems die Programmiersprache Java [63] [64] als eine rein objektorientierte und plattformunabhängige Sprache vorgestellt. Die Plattformunabhängigkeit von Java wird dadurch erreicht, dass der Java-Compiler den Sourcecode nicht in Maschinencode, sondern in Bytecode umwandelt, der von der Java-Virtual-Machine (JVM) ausgeführt wird. Der Bytecode kann ohne eine aufwendige Portierung auf anderen Plattformen ausgeführt werden, auf denen eine JVM zur Verfügung steht. Weitere Vorteile von Java gegenüber hybriden Programmiersprachen wie C++ [66] bestehen darin, dass stets eine strenge Typprüfung durchgeführt und die fehleranfällige Pointerarithmetik von C++ ausgeschlossen wird. Im Rahmen dieser Arbeit wurden sämtliche Implementierungen in der Programmiersprache Java vorgenommen.

4.3.2 Eclipse

Die integrierte Entwicklungsumgebung (Integrated Development Environment - IDE) Eclipse [67] wird im Rahmen dieser Arbeit für die Entwicklung und Implementierung der Basisarchitektur, aber auch für die Erstellung der Softwareprojekte aus Kapitel 5 verwendet. Die IDE wurde ursprünglich von IBM unter dem Namen Visual Age for Java entwickelt und im Jahr 2001 als Open-Source-Framework freigegeben. Seit 2004 steht die IDE unter dem Dach der Eclipse Foundation, welche seitdem für die Weiterentwicklung verantwortlich ist. Neben der Programmiersprache Java unterstützt Eclipse eine Vielzahl von anderen Programmiersprachen.

Eclipse basiert selbst auf der Programmiersprache Java und die Architektur der IDE, die auf der Eclipse Rich Client Platform [68] aufbaut, kann durch Plugins den jeweiligen Anforderungen angepasst werden. Für die im Rahmen dieser Arbeit verwendeten Frameworks existieren entsprechende Plugins.

4.4 Framework für die Logikschicht

Es wurde ein Java-Framework für Web-Anwendungen gesucht, das geeignet ist, die funktionellen Anforderungen an die Logikschicht zu erfüllen. Besonderes Augenmerk wurde darauf gelegt, dass das Framework Anwendungen in Form eigener Komponenten einbinden kann, Unterstützung bei der Umsetzung der Zugriffssteuerungen des RBVC-Modells bietet und eine saubere Trennung von Anwendungen und der Sichtengenerierung ermöglicht.

Das Web-Development-Framework Apache Cocoon [69]-[73] erfüllt die vorher genannten Kriterien sehr gut und es wurde entschieden, dieses als Basis für die komponentenbasierte Entwicklung der Web-Anwendungen unter Anwendung des RBVC-Modells zu verwenden. Apache Cocoon ist ein Open-Source Framework, das von der Apache Software Foundation herausgegeben wird.

In den folgenden Unterkapiteln werden die einzelnen Elemente von Apache Cocoon beschrieben, die für die Realisierung des RBVC-Modells für Web-Anwendungen notwendig sind.

4.4.1 Separation of Concerns

Unter *Separation of Concerns* (SoC) [74] [75] versteht man das Aufteilen eines Programms in klar voneinander trennbare funktionelle Bereiche (Abb. 24), deren Funktionen sich so wenig wie möglich überlappen sollten.

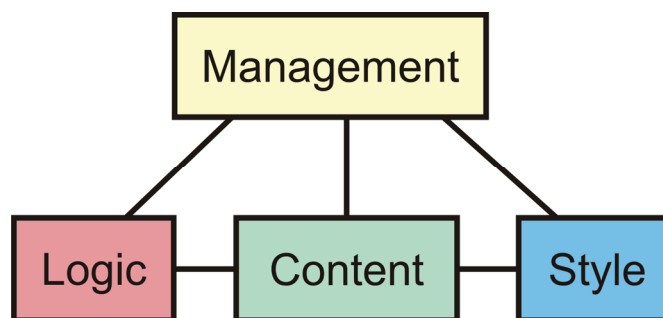


Abb. 24: Separation of Concerns [76]

4.4.2 Servlet

Apache Cocoon selbst ist ein komplexes Java Servlet [77]-[79], das in einem Web-Container [80] läuft. Web-Container sind Programme, in denen Web-Anwendungen betrieben werden. Apache Cocoon liefert Jetty [81] als Web-Container mit, der neben Apache Tomcat [82] einen der bekanntesten Web-Container darstellt. Prinzipiell lässt sich Apache Cocoon aber auch in jedem anderen Web-Container zur Verfügung stellen.

Web-Container werden als Bestandteil von *Java 2 Platform Enterprise Edition* (J2EE)-Application-Servern eingesetzt, so dass Apache Cocoon auch in einer solchen Umgebung eingesetzt werden kann.

Für die Umsetzung der Basisarchitektur wurde Jetty als Web-Container eingesetzt.

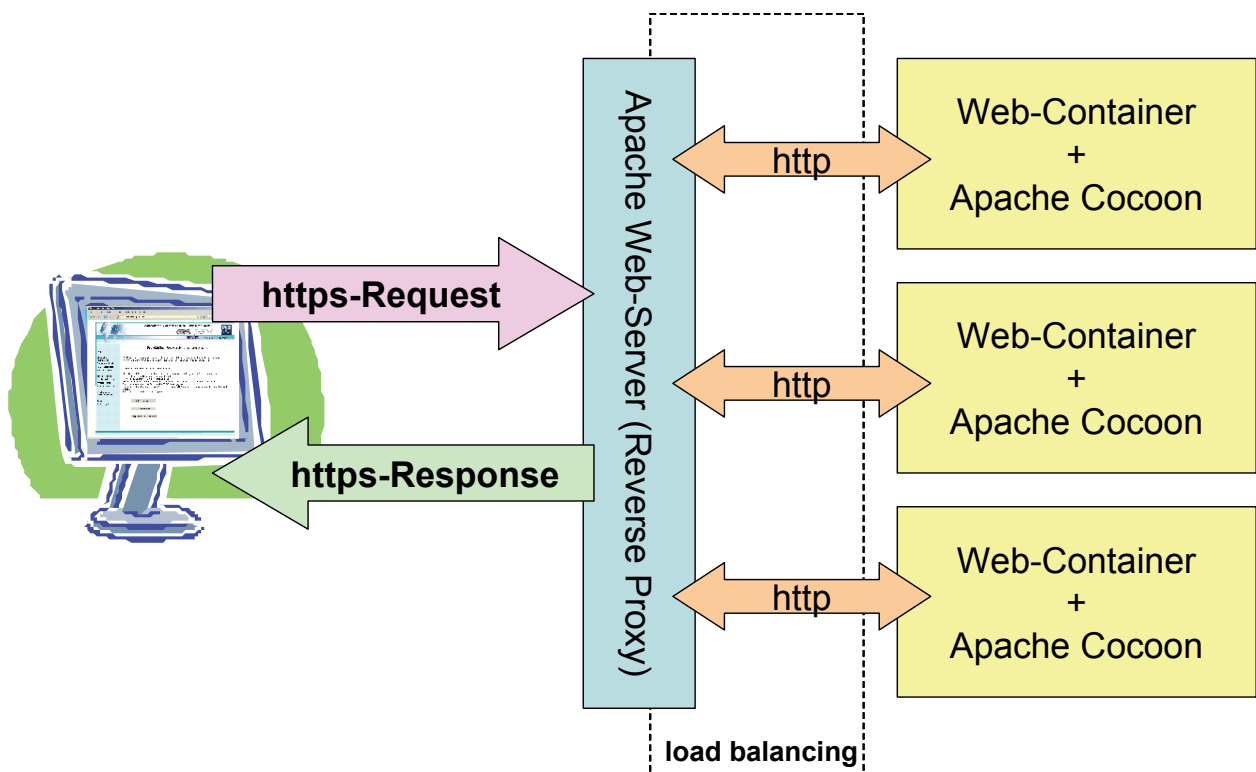


Abb. 25: Apache Web-Server als Reverse-Proxy-Server mit Lastverteilung

Der Web-Container Jetty wird nicht direkt an das Internet angebunden, sondern über einen zwischengeschalteten Apache Web-Server (Abb. 25), der als Reverse-Proxy-Server [83] [84] eingesetzt wird. Dieses Vorgehen bietet mehrere Vorteile:

- Statische Elemente, wie beispielsweise Bilder und große Downloads, können direkt vom Reverse-Proxy-Server geladen werden, was die Übertragungsgeschwindigkeit signifikant erhöht.
- Statische Inhalte, die vom Web-Container stammen, können vom Reverse-Proxy-Server zwischengespeichert werden, dadurch werden die dahinter liegenden Anwendungen weniger belastet.
- Es kann eine Lastverteilung (load balancing) auf mehrere Web-Container stattfinden.
- Die rechenintensive Verschlüsselung der Verbindungen mit SSL findet nur zwischen dem Reverse-Proxy-Server und dem Browser statt. Dadurch werden die Server, auf denen die Web-Container und Apache Cocoon ausgeführt werden, weniger belastet.

- Falls eine Web-Anwendung nicht erreichbar ist, können Fehlermeldungen für den Benutzer ausgegeben werden.

4.4.3 Control Flow mit Continuations

Traditionell bestehen Web-Anwendungen aus einer Aneinanderreihung von unterschiedlichen Zuständen. Die Verbindung zwischen Klient und Server über http und https ist zustandslos. Da aber die Anwendung gleichzeitig nur einen Zustand einnehmen kann, verändert jede Anfrage des Klienten den Zustand des dahinter stehenden *endlichen Automaten*³⁵ (EA), der in einen anderen Zustand übergeht. Diese Zustandsänderung kann unterschiedliche Bereiche betreffen und unter anderem auch Datenbankänderungen mit einbeziehen. Dieses traditionelle Modell für Web-Anwendungen funktioniert sehr gut, solange es sich um kleine Anwendungen handelt, die nur wenige Zustände und Zustandsübergänge haben.

Wenn Anwendungen komplexer werden, stößt man sehr schnell an die Grenzen des EA. Deshalb wurde in Apache Cocoon der sogenannte *Control Flow* mit Continuations [85] eingeführt, bei dem sich die Web-Anwendung so verhält, als wäre sie wie eine normale Anwendung modelliert, die direkt auf einem Klienten ausgeführt wird.

Eine Anfrage startet einen *Control Flow*. Dieser lässt von der Web-Anwendung eine Operation ausführen und wird dadurch in einen neuen inneren Zustand übergeführt. Dieser Zustand wird mit einer Continuation-ID versehen, von Cocoon gespeichert und im Rahmen der Antwort an den Klienten übermittelt. Wird nun eine erneute Anfrage unter Angabe der Continuation-ID gestellt, dann setzt der *Control Flow* seine Arbeit an der Stelle fort, an der die letzte Antwort gegeben wurde. Alle inneren Zustände bleiben erhalten.

4.4.4 Cocoon Forms

Cocoon Forms (CForms) [86] ist ein Formular Framework, das von Apache Cocoon unter Verwendung des *Control Flow* zur Verfügung gestellt wird, um komplexe Formulareingaben zu vereinfachen. Für jedes Formular müssen mindestens die beiden Dateien *Model* und *Template* erzeugt werden. Die weiteren Dateien sind optional.

³⁵ In der Literatur wird hierfür auch sehr häufig die englische Bezeichnung *Finite State Machine* (FSM) verwendet.

1. Model: Diese Datei definiert die Formularelemente (Widgets) und deren Eigenschaften, Typen, etc.
2. Template: Diese Datei regelt die Anordnung der Formularelemente auf einer Seite.
3. Binding: Mit dieser optionalen Datei können die Formularinhalte direkt an Java-Bean-Objekte³⁶ gebunden werden.
4. I18n³⁷: Diese optionale Datei regelt die Internationalisierung der Formulare.

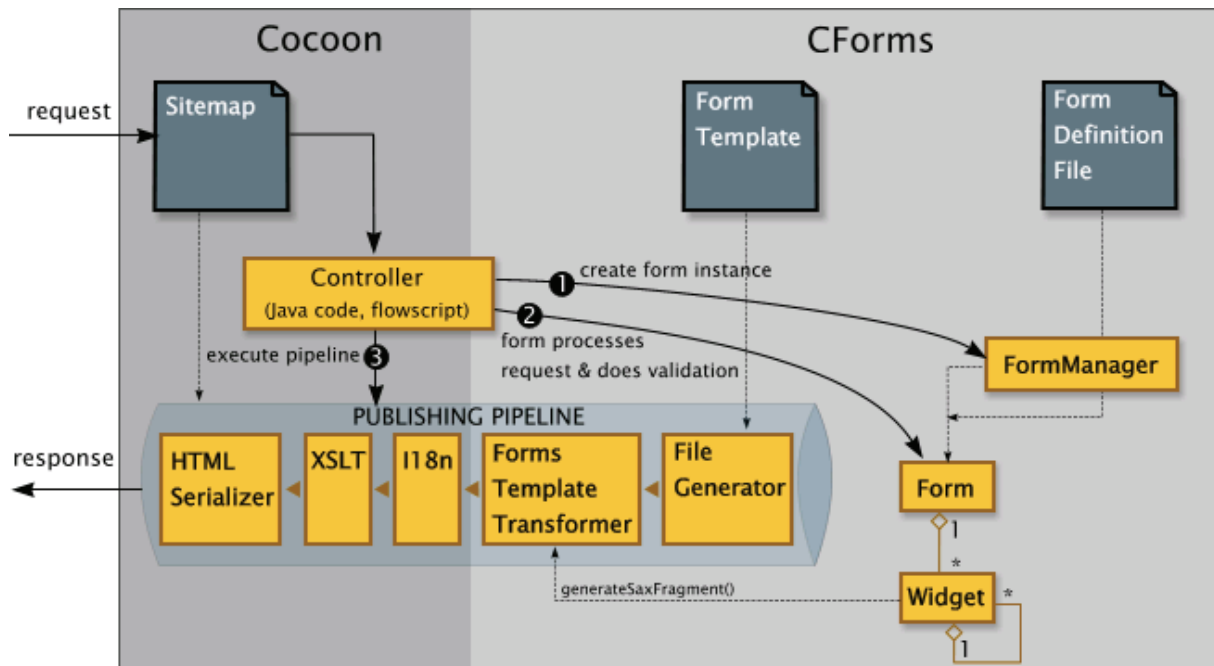


Abb. 26: Ablauf einer Formulareinstellung mit Cocoon Forms [86]

Abb. 26 stellt ein mögliches Szenario für den Ablauf einer Formulareinstellung dar. Eine Anfrage (request) wird von Apache Cocoon durch die Konfiguration in der Sitemap (vgl. Kap. 4.4.5) an den entsprechenden *Control Flow* übergeben. Dieser nutzt die Cocoon-Komponente *FormManager* (1) um mit Hilfe der Model-Datei (form file definition) ein Formular (2) aus beliebig vielen geschachtelten Widgets zu generieren. Die Veröffentlichungs-Pipeline (publishing pipeline) wird vom *Control Flow* gestartet (3) und generiert (file generator) einen XML-Strom aus der Template-Datei (form template) des Formulars. In der nächsten Stufe der Pipeline wandelt der *Forms-Template-Transformer* den Template-XML-Strom zusammen mit den Widget-

³⁶ Java-Beans sind serialisierbare Klassen, auf die über öffentliche Zugriffsmethoden (sogenannte getter und setter) zugegriffen werden kann.

³⁷ Internationalisierung

Informationen, die als SAX³⁸-Fragmente enthalten sind, in einen Formular-XML-Strom um. Dieser wird in den weiteren Stufen in die gewünschte Sprache übersetzt (I18n), mit dem Styling der Formulare versehen (XSLT) und danach vom HTML-Serializer als HTML-Seite (response) an den Klienten übermittelt.

CForms bietet die klientenseitige Validierung von Eingabewerten an. Dazu können bei der Widget-Definition einschränkende Datentypen, Bereiche und reguläre Ausdrücke angegeben werden, deren Einhaltung vor Absendung des Formulars im Browser mit JavaScript überprüft wird.

Eine Erweiterung von CForms stellt die asynchrone Kommunikation zwischen Browser und Server über AJAX-Funktionen (vgl. Kap. 4.2.1) zur Verfügung.

4.4.5 Sitemaps und Pipelines

In den Sitemaps wird die gesamte Seitenstruktur einer oder auch mehrerer Webanwendungen hierarchisch abgebildet. Die Sitemap-Dateien in Apache Cocoon bestehen aus XML-Dateien, in denen die Seitenstruktur konfiguriert werden kann. Auf höchster Ebene existiert eine Sitemap-Datei als globaler Einstiegspunkt, über die an beliebig viele Stufen von Unter-Sitemaps verwiesen werden kann. Somit kann für jede Web-Anwendung, die unter einer Instanz des Frameworks läuft, in einer Unter-Sitemap eine eigene Konfiguration vorgenommen werden.

In einer Sitemap-Datei werden in Abhängigkeit von der aufrufenden URL³⁹ durch einen Mustervergleich unterschiedliche Pipelines aufgerufen. Innerhalb der Pipelines wird ein XML-Strom basierend auf der *Simple API for XML* (SAX) [87] zwischen den einzelnen Stufen weitergereicht.

In Abb. 27 ist der Ablauf eines URL-Aufrufes [88], der für die Implementierung der Basisarchitektur konzipiert wurde, dargestellt. Wenn das Muster der aufrufenden URL mit der Konfiguration der Sitemap übereinstimmt, beginnt an dieser Stelle die Flow-Pipeline mit einer Action. In dieser Action wird im ersten Schritt die RBVC-Zugriffssteuerung aufgerufen. Diese stellt in einem ersten Schritt fest, ob der aktuelle Benutzer bereits authentifiziert ist. Ist dies nicht der Fall, dann wird zur Authentifizie-

³⁸ *Simple API for XML*

³⁹ Die Spezifikation des *Unified Resource Locators* (URL) wurde ersetzt durch die neuere Spezifikation des *Unified Resource Identifier* (URI). In der Literatur werden die beiden Begriffe synonym verwendet, der Begriff URL kommt aber häufiger vor. Deswegen wird in dieser Arbeit auch der Begriff URL verwendet, obwohl die Spezifikation des URI gemeint ist.

rung weitergeleitet und eine weitere Ausführung der Pipeline beendet. Ist die Überprüfung der Identität des Benutzers bereits erfolgt, so wird er im nächsten Schritt autorisiert. Dabei wird festgestellt, ob er die notwendige Rolle zugewiesen hat, um die durch die URL angeforderte Berechtigung zu erhalten. Ergibt die Prüfung, dass der Benutzer nicht zugriffsberechtigt ist, so wird dies an die Fehler-Pipeline weitergeleitet, die eine entsprechende Fehlermeldung ausgibt. Ist der Benutzer zugriffsberechtigt, so wird der entsprechende Control-Flow aufgerufen. In der konkreten Implementierung ist dies eine Java-Klasse, die den Flow durch die Web-Anwendung steuert.

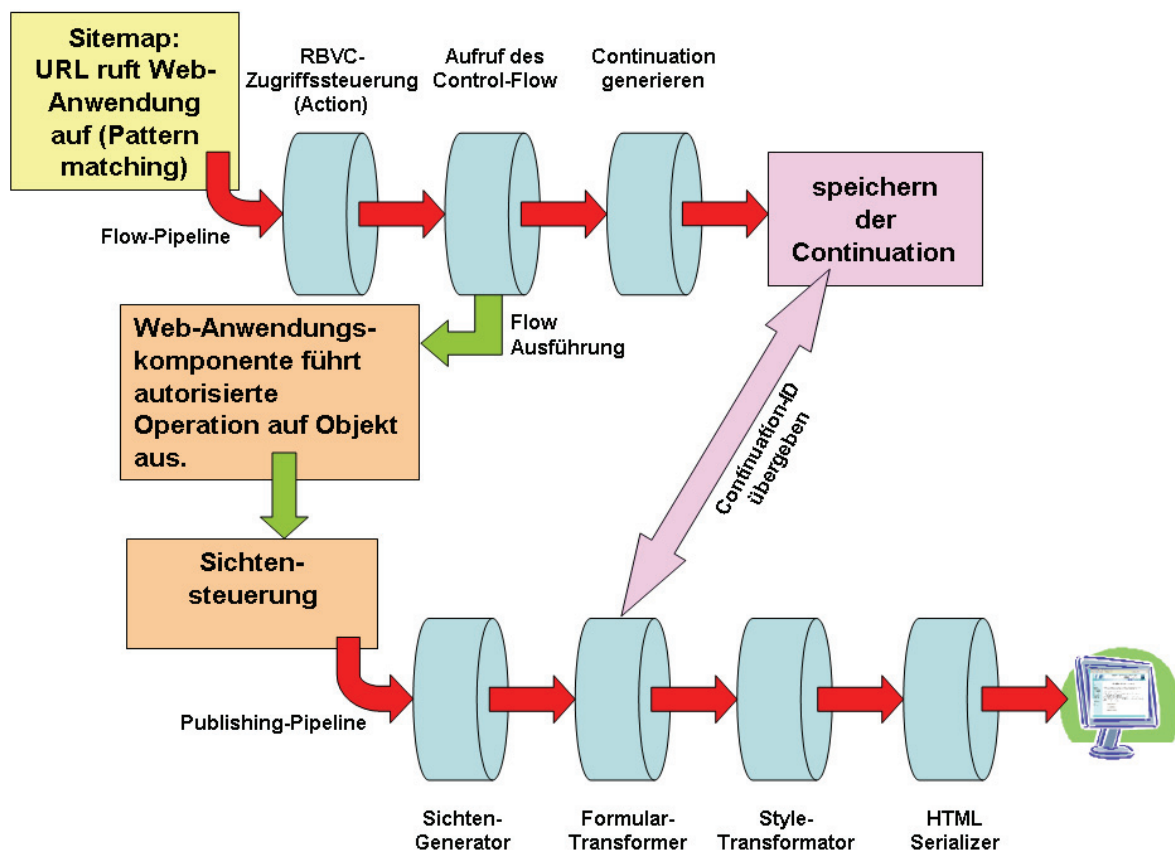


Abb. 27: Pipeline verarbeitet einen Control Flow-Aufruf durch eine URL

Der Control Flow wird nun ausgeführt und ruft die Komponente der Web-Anwendung auf. Diese führt eine Operation, die auf die Datenschicht zurückgreifen kann, eigenständig durch. Wenn die Operation abgeschlossen ist, übergibt die Web-Anwendung die Daten für die Visualisierung an die Sichtensteuerung, welche resultierend aus der aktivierten Rolle und der damit verbundenen Sicht des Benutzers die Publishing-Pipeline anstößt. Die Flow-Pipeline wird nun beendet und der Zustand des Flows wird unter der Continuation-ID abgespeichert.

Die Publishing-Pipeline beginnt mit dem Sichten-Generator, der die Daten aus der Web-Anwendung rollenoptimiert aufbereitet. Der Formular-Transformator erstellt ein CForms-Formular und bindet die nächste Continuation-ID als Einsprungstelle für weitere Flow-Aufrufe in das Formular ein. Im nächsten Schritt wird im Style-Transformator das Aussehen mit dem Formular verbunden und danach durch den HTML-Serializer ausgegeben und an den Browser zur Darstellung übermittelt.

Wenn der Benutzer im Browser seine Arbeit fortsetzt und eine neue Server-Interaktion anstößt, dann wird im Rahmen dieser neuen Anfrage eine URL mit der im Formular eingebundenen Continuation-ID an den Server übermittelt. Die Anfrage wird serverseitig ausgewertet und die entsprechende Flow-Pipeline gestartet. Der weitere Vorgang ist identisch mit dem bereits beschriebenen ersten URL-Aufruf (Abb. 27), mit dem Unterschied, dass jetzt eine Fortführung des Flow-Control anhand des vorher gespeicherten Zustandes stattfindet.

4.4.6 Komponenten in Apache Cocoon

Apache Cocoon kann durch eigene Komponenten erweitert werden. Hierzu bietet das Framework eine Schnittstelle an, die auf dem Entwurfsmuster *Inversion of Control* (IoC) basiert.

IoC⁴⁰ [89] ist ein Ansatz, durch dessen Hilfe die normal existierende Kopplung innerhalb eines Programms reduziert werden soll. Die Umsetzung des Entwurfsmusters in Apache Cocoon geschieht dadurch, dass das Avalon Framework als IoC-Container verwendet wird, der selbst ein Bestandteil von Apache Excalibur [90] ist.

Wenn in Abb. 28 das bestehende Java-Objekt (Object A) neue Exemplare der Klasse B und C in Form von Objekten B und C (Object B, Object C) instanziiert, dann sind diese neuen Exemplare fest mit dem Objekt A gekoppelt.

Im Gegensatz dazu sind die Java-Objekte Object B und Object C lose mit Object A gekoppelt (Abb. 29), wenn diese beispielsweise im Konstruktor der Klasse A als Interfaces übergeben und innerhalb der Klasse A nur diese Interfaces verwendet werden.

⁴⁰ In der Literatur ist in diesem Zusammenhang auch häufig die Rede vom *Hollywood-Prinzip*: „don't call us, we'll call you“ [90]

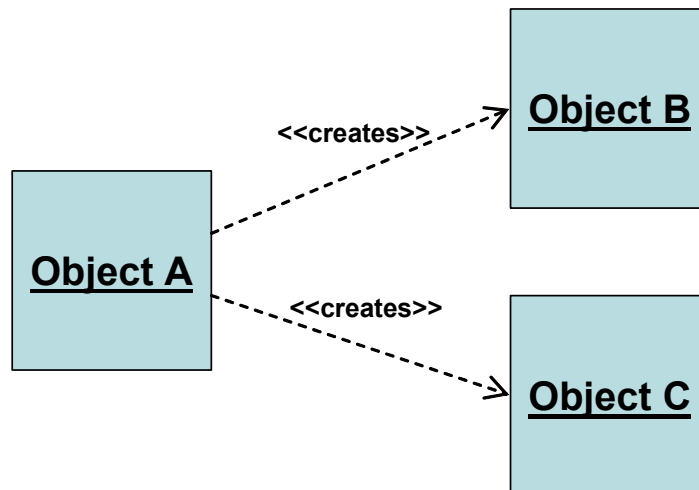


Abb. 28: Feste Kopplung ohne Inversion of Control (vgl. [91])

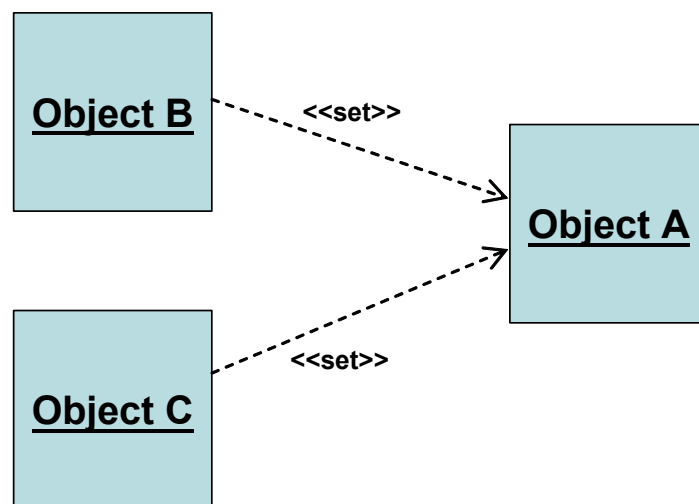


Abb. 29: Lose Kopplung durch Inversion of Control (vgl. [91])

Die im Rahmen dieser Arbeit entwickelten Manager-Klassen⁴¹ des Rollenmodells werden jeweils als eigenständige Avalon-Komponenten implementiert (Kap. 4.7). Dieses Vorgehen gewährleistet, dass einzelne Komponenten problemlos ausgetauscht und durch andere ersetzt werden können, sofern die Manager-Interfaces⁴², welche die Komponentenschnittstellen sind, eingehalten werden.

⁴¹ Analog zur Namenskonvention in Apache Cocoon werden mit dem Begriff *Manager-Klassen* die Implementierungen der Komponenten bezeichnet.

⁴² Analog zur Namenskonvention in Apache Cocoon werden mit dem Begriff *Manager-Interfaces* die Implementierungen der Komponentenschnittstellen bezeichnet.

4.5 Optionale Ergänzung der Logikschicht durch Google Web Toolkit

Toolkit

Das Google Web Toolkit (GWT) [92]-[94] stellt eine AJAX-basierte Alternative zur Verwendung der in Kap. 4.4.4 beschriebenen Cocoon Forms dar. Bei GWT handelt es sich um eine Java-Entwicklungsumgebung, die im Mai 2006 von Google veröffentlicht wurde und seit Version 1.3 unter die Apache 2.0 Open Source Lizenz [95] gestellt wurde und damit frei verfügbar ist.

Der Vorteil von GWT liegt darin, dass eine Programmierung der Oberfläche direkt in Java erfolgen kann. Eine Umsetzung des Java-Codes in äquivalenten JavaScript-Code, der auf dem Browser ausgeführt werden kann, wird durch den GWT-Compiler durchgeführt. GWT unterstützt alle gängigen Browser wie beispielsweise Internet Explorer, Mozilla Firefox und Safari [96]. Es findet durch GWT eine Trennung des Java-Codes, der für die Entwicklung der Oberfläche erstellt wird, vom GWT-compilerisierten JavaScript-Code, der im Browser ausgeführt wird, statt. Die Entwickler müssen sich daher nicht mehr mit Browser-Inkompatibilitäten beschäftigen.

Während der Entwicklung wird GWT im sogenannten *Hosted Mode* [96] betrieben, der spezielle Unterstützung für die Fehlersuche enthält. Ist eine Oberfläche fertig gestellt, wird die Oberfläche in den *Web Mode* versetzt, um sie zu veröffentlichen.

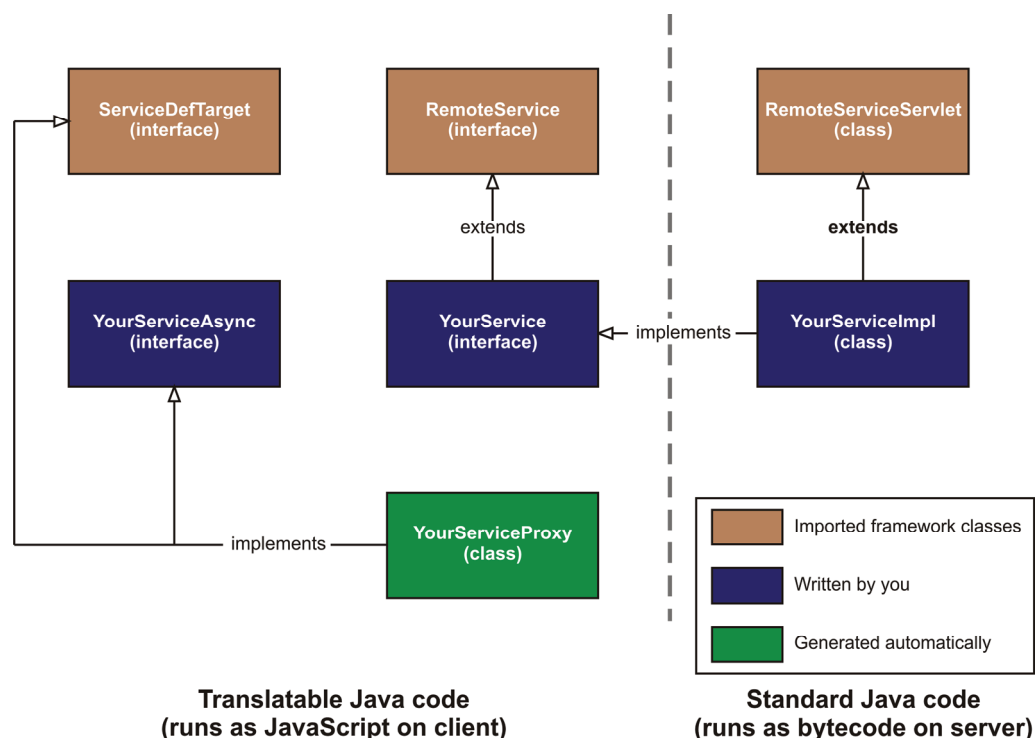


Abb. 30: Aufbau der Google Web Toolkit Remote-Procedure-Calls [96]

Die Kommunikation der Oberfläche mit dem Server findet durch *Remote-Procedure-Calls* (RPC) statt. In Abb. 30 ist im linken Bereich der in JavaScript übersetzte Java-Code dargestellt, der im Browser läuft. Im rechten Bereich ist der serverseitige Java-Code dargestellt, der als Servlet im Web-Container läuft und die Remote-Procedure-Calls entgegennimmt.

Die klientenseitige Programmierung der Remote-Procedure-Calls erfolgt durch Implementierung der blau dargestellten Schnittstellen *YourServiceAsync* und *YourService*, welche die Schnittstelle *RemoteService* des Frameworks erweitern. *YourService* ist die Schnittstelle für die synchrone Kommunikation mit dem Server. Die Methoden, die hierin definiert werden, erhalten direkt Rückgabewerte. Die Schnittstelle *YourServiceAsync* dient der asynchronen Kommunikation, deren Methoden keine Rückgabewerte, sondern einen Rückruf-Parameter enthalten, der aufgerufen wird, sobald die asynchrone Kommunikation beendet wurde. Die grün dargestellte Klasse *YourServiceProxy* implementiert sowohl die eigene Schnittstelle *YourServiceAsync* als auch die vom Framework bereitgestellte Schnittstelle *ServiceDefTarget*, um das Ziel der Remote-Aufrufe festzulegen. Sie wird durch den GWT-Compiler automatisch generiert.

Serverseitig muss die Klasse *YourServiceImpl* programmiert werden, welche die Klasse *RemoteServiceServlet* des Frameworks erweitert und dadurch ein Java Servlet implementiert. Weiter muss die Klasse *YourServiceImpl* die klientenseitigen Schnittstellen implementieren, um die Remote-Procedure-Calls ausführen zu können.

4.6 Realisierung der Datenschicht

Die Datenschicht kann prinzipiell aus vielen unterschiedlichen Datenquellen bestehen, die sich je nach Projekt, in dem die Basisarchitektur eingesetzt werden soll, unterscheiden können. In diesem Kapitel wird auf den kleinsten gemeinsamen Nenner bei der Realisierung der Datenschicht eingegangen. Dieser besteht aus einer relationalen Datenbank, die alle persistenten Daten des RBVC-Modells speichert.

Werden weitere Datenquellen benötigt, so können diese als neue Komponenten bereitgestellt werden, sofern eine dafür vorgegebene Schnittstelle für Datenquellen implementiert wird.

4.6.1 Persistenz-Framework

Aus den Rahmenbedingungen ergibt sich, dass die Implementierung der Basisarchitektur von den Datenbankdialekten entkoppelt sein soll. Deswegen wird eine Entkopplung auf Basis eines Persistenz-Frameworks durchgeführt (vgl. 4.2.3). Für die Programmiersprache Java gibt es das Open-Source-Persistenz-Framework Hibernate [97]-[99], das als Datenquellen-Komponente in Apache Cocoon eingebunden wird.

Hibernate stellt der Web-Anwendung sogenannte POJO⁴³s zur Verfügung, über die auf die Datenbankinhalte zugegriffen werden kann. Die Abfrage persistenter Objekte erfolgt bei Hibernate über die eigene Abfragesprache *Hibernate Query Language* (HQL), die eine große Ähnlichkeit zu SQL besitzt, oder über eine objektorientierte API⁴⁴, die auf Kriterien beruht.

4.6.2 Relationale Datenbank

Bei den Implementierungen, die auf der Basisarchitektur aufbauen, wird die frei verfügbare Datenbank MySQL eingesetzt, diese lässt sich aber prinzipiell durch jede andere Datenbank ersetzen, deren SQL-Dialekt vom Persistenz-Framework Hibernate unterstützt wird.

Datenbankseitig werden, um die Transaktionssicherheit für konkurrierende Zugriffe zu gewähren, MySQL-InnoDB-Tabellen verwendet, welche die Definition von Relationen zwischen den Tabellen zulassen.

4.7 Implementierung der Komponenten der Basisarchitektur

In diesem Unterkapitel werden die Schnittstellen der zentralen Komponentenschnittstellen und deren Standard-Implementierungen für die Basisarchitektur beschrieben. Unter Apache Cocoon (vgl. 4.4) werden die Implementierungen der Komponenten in einer speziellen XML-Datei konfiguriert und somit dem IoC-Container (vgl. 4.4.6) zur Verfügung gestellt. Eine Abhängigkeit zwischen Komponenten in Form einer Beziehung wird stets über den IoC-Container abgebildet, so dass auch hier einzelne Komponenten problemlos ausgetauscht werden können.

⁴³ *Plain Old Java Object* (POJO) ist ein gängiger Begriff für einfache Java-Objekte, deren Member-Variablen über Getter- und Setter-Methoden gelesen und gesetzt werden können.

⁴⁴ Application Programming Interface

4.7.1 RbacManager

Die Komponentenschnittstelle mit dem Namen RbacManager stellt die Methoden für die Zugriffssteuerung des RBVC-Modells bereit. Sobald ein Benutzer am System authentifiziert (vgl. 4.7.2) ist, wird eine Session für ihn angelegt, in der er seine Transaktionen in Form von Operationen auf Objekte durchführen kann. Bevor eine Transaktion durchgeführt wird, muss die hierfür notwendige Berechtigung überprüft werden. Im System kann diese Anfrage jederzeit gestellt werden. In ihr wird überprüft, ob ein Benutzer aufgrund seiner Rollen im System und unter der Berücksichtigung der Kriterien der dynamischen Aufgabenteilung⁴⁵ (DSD) dazu berechtigt ist, eine Operation auf ein Objekt durchzuführen. Ist er dazu berechtigt und möchte er diese Operation ausführen, so wird ihm ein Ticket ausgestellt, das eine festgelegte maximale Lebensdauer⁴⁶ hat. Die Lebensdauerbegrenzung ist deswegen notwendig, damit das System im Fehlerfall nicht dauerhaft gesperrt wird. Innerhalb der Lebensdauer eines Tickets kann für dasselbe Objekt kein neues Ticket ausgestellt werden, da ansonsten DSD-Regeln missachtet werden könnten.

Im Regelfall wird das Ticket innerhalb der Lebensdauer im Rahmen einer Operation eingelöst. Hierzu werden die Daten der Transaktion persistent gespeichert, so dass diese für zukünftige DSD-Überprüfungen zur Verfügung stehen.

4.7.2 AuthenticationManager

Die Komponentenschnittstelle AuthenticationManager bietet die Möglichkeit, unterschiedliche Arten der Authentifizierung (vgl. Kap. 3.6) an die Basisarchitektur anzubinden. Die Standard-Implementierung verwendet die sehr häufig eingesetzte Methode bestehend aus Benutzernamen und Passwort. Andere Implementierungen erlauben beliebige Arten und Kombinationen der Authentifizierungsmethoden.

Die Authentifizierung über Web-Service-Schnittstellen findet ebenfalls über die Komponentenschnittstelle AuthenticationManager statt, so dass hier eine allgemeine Authentifizierungsmöglichkeit geschaffen wurde. Wird ein Ticket innerhalb seiner Lebensdauer nicht eingelöst, wird es kommentarlos gelöscht.

⁴⁵ vgl. Kapitel 2.2.4

⁴⁶ In der Praxis sind dies einige Millisekunden

4.7.3 MenueManager

Die Komponentenschnittstelle MenueManager wird für einen Teil der Umsetzung des RBVC-Modells benötigt, um die entsprechenden Menüs für die Benutzer zu generieren, die genau die Sichten der Funktionen zur Auswahl anzeigen, für die sie durch ihre Rollen autorisiert sind. Ist ein Benutzer in mehreren Rollen in unterschiedlich ausgeprägten Funktionen in Form von dazugehörigen Sichten für denselben Menüpunkt autorisiert, so ist der MenueManager dafür verantwortlich, die notwendige Auflösung des Konfliktes durchzuführen (vgl. 3.4.2).

4.7.4 ViewControlManager und ViewGenerationManager

Der ViewControlManager ist die zentrale Komponentenschnittstelle zur RBVC-Sichtensteuerung. Wählt ein Benutzer im Menü, das vom MenueManager erzeugt wurde, eine rollenabhängige Funktion aus, so wird im kollisionsfreien Fall automatisch vom ViewGenerationManager die entsprechende Darstellungskomponente angesprochen. Im Fall einer Sichtenkollision muss die Rolle, in der die Funktion ausgeführt werden soll, im Menü manuell vom Benutzer ausgewählt werden (vgl. 3.4.2).

4.7.5 SourceManager

Die Komponentenschnittstelle SourceManager dient zur Anbindung unterschiedlicher lokaler Datenquellen. Sie ist die Schnittstelle, welche die Logikschicht mit der Datenschicht verbindet. Es gibt Standardimplementierungen für die gebräuchlichsten Datenquellen (Datenbanken, Dateien, etc.), es können aber prinzipiell beliebige Datenquellen angebunden werden, sofern diese Schnittstelle erweitert wird. Auch entfernte Datenquellen, die über den RemoteDataObjectManager (vgl. 4.7.7) angesprochen werden, erweitern diese Schnittstelle.

Die angebundenen Datenquellen müssen, sofern darin mehrere Objekte liegen, über ein eindeutiges Merkmal identifiziert werden können, so dass über RBAC der Zugriff darauf kontrolliert werden kann.

4.7.6 RemoteAuthorizationManager und RemoteAccessManager

Die Komponentenschnittstelle RemoteAuthorizationManager und deren Erweiterung RemoteAccessManager dienen zur Anbindung räumlich getrennter Systeme über das Internet. Die entfernten Systeme können beispielsweise bei anderen Organisati-

onen betrieben werden und erlauben den präsentationsseitigen Zugriff auf ein System, das auf der RBVC-Basisarchitektur aufbaut.

Der RemoteAuthorizationManager bietet nur die Möglichkeit einer räumlich getrennten und systemübergreifenden Nutzung der Zugriffssteuerung bezogen auf die reine Autorisierung über eine Web-Service-Schnittstelle (Abb. 31). Der RemoteAccessManager erweitert diese Funktion und erlaubt eine entfernte Nutzung aller Systemfunktionen, sofern dafür eine spezielle Web-Service-Sicht erstellt wurde.

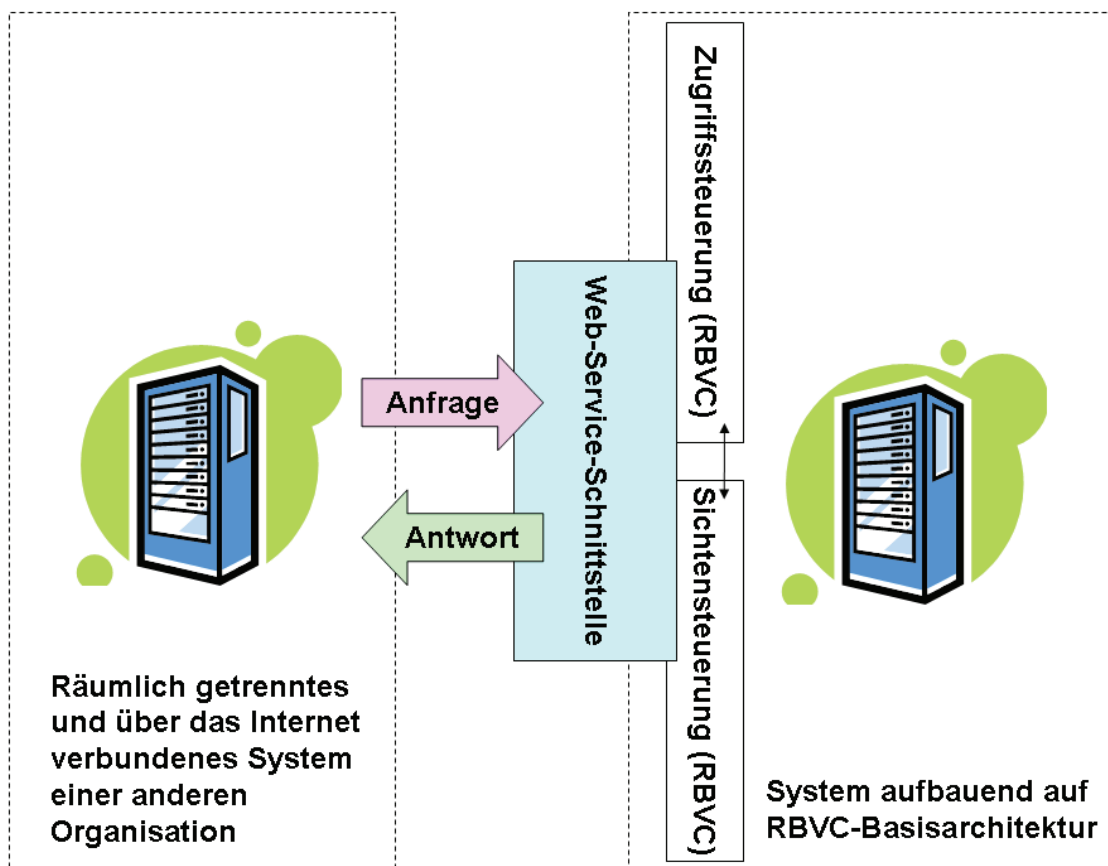


Abb. 31: Zugriff eines räumlich getrennten Systems über das Internet

4.7.7 RemoteApplicationManager und RemoteDataObjectManager

Die Komponentenschnittstelle RemoteApplicationManager dient zur Anbindung von Applikationen, die Schnittstelle RemoteDataObjectManager zur Anbindung von Datenobjekten. Sie können beispielsweise bei anderen Organisationen betrieben werden und erlauben den Zugriff der RBVC-Basisarchitektur über eine Web-Service-Schnittstelle.

In Abb. 32 ist dargestellt, wie die Implementierung der Komponentenschnittstelle RemoteApplicationManager, dargestellt in *Anwendung 1*, zweigeteilt ist. Die Aufteilung geschieht über das Strukturmuster Stellvertreter (Proxy) [100]. In der Logikschicht verbleibt der Proxy der *Anwendung 1*, der über die Web-Service-Schnittstelle und damit über das Internet auf die implementierende *Remote-Anwendung 1* zugreift. Weiter ist der Zugriff der *Anwendung 2* unter Verwendung des RemoteDataObjectManagers über eine Web-Service-Schnittstelle auf Remote-Datenobjekte dargestellt.

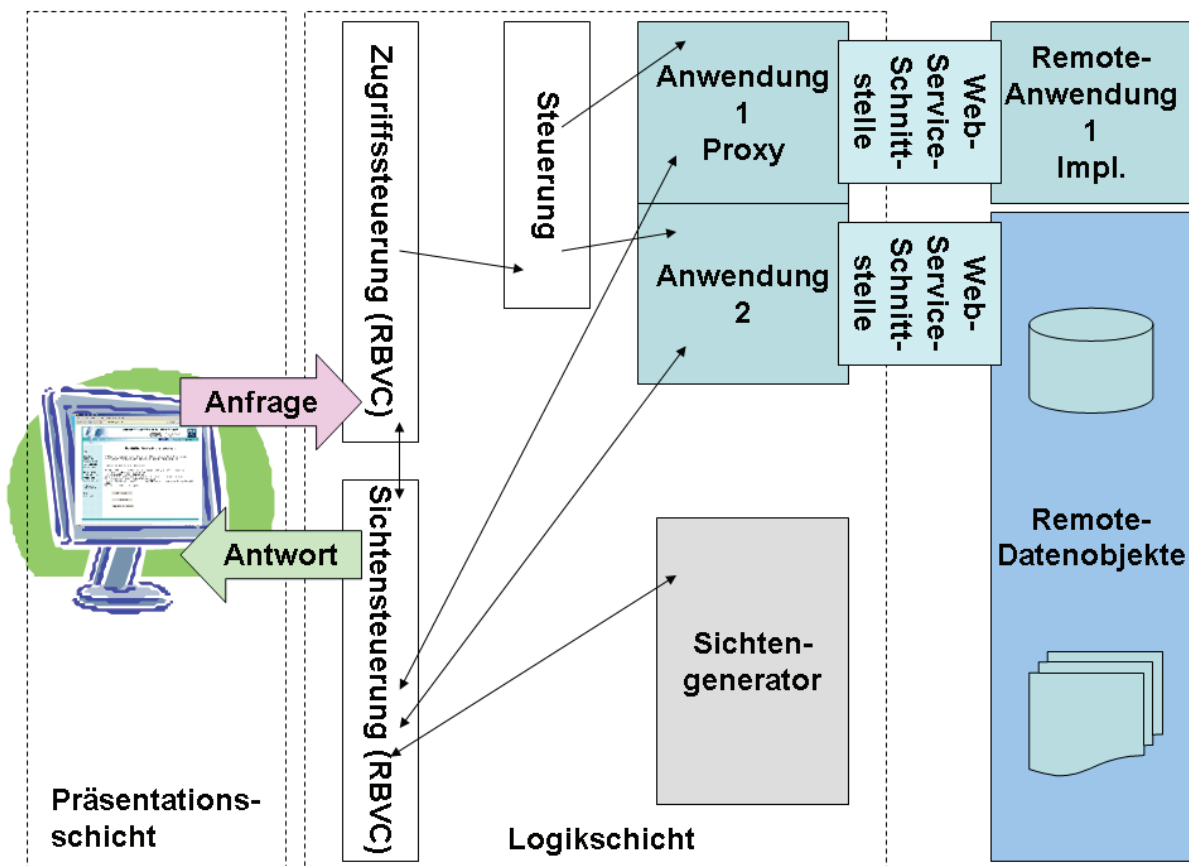


Abb. 32: Zugriff auf räumlich getrennte Anwendungen und Datenobjekte über das Internet

4.7.8 Test der Komponenten

Im Rahmen der testgetriebenen Entwicklung von Komponenten wurden konsequent JUnit-Tests [101] [102] eingesetzt, welche automatisierte Tests einzelner Einheiten erlauben. Um sicherzustellen, dass die Testfälle nicht nur einer bestehenden Komponentenimplementierung angepasst werden, werden diese vor Erstellung der Komponente für die Komponentenschnittstelle erstellt. Dadurch wird verhindert, dass Testfälle übersehen werden.

Auch bei Änderungen im Rahmen des Refactorings werden die Tests vorher aufgerufen, um sicherzustellen, dass die Software vor der Änderung fehlerfrei lief. Nach der Änderung werden die Tests erneut gestartet und man kann anhand des Testergebnisses feststellen, ob das Refactoring erfolgreich oder fehlerhaft war. Im zweiten Fall muss solange nachgebessert werden, bis die Tests fehlerfrei ablaufen.

5 Anwendungen des Rollenmodells

Um die Tragfähigkeit des entwickelten Rollenmodells für den Fachbereich Kerntechnik nachzuweisen, wurden die einzelnen Iterationsstufen der RBVC-Modellentwicklung in Softwareprojekten angewandt und umgesetzt. Durch dieses Vorgehen konnten einerseits neue Anforderungen an das Rollenmodell direkt aus den Erfahrungen bei der Implementierung abgeleitet werden, andererseits konnte eine Systemarchitektur entwickelt werden, auf deren Basis alle im Folgenden vorgestellten Produktionssysteme aufgebaut werden konnten.

5.1 Fallbeispiel Wkind und Integration in eine Lehr-/Lernumgebung

Das Programm Wkind, Fallbeispiel für Simulationsprogramme aus Kapitel 1.6.1, wurde im Rahmen eines europäischen Kurses mit dem Themenschwerpunkt Hochtemperaturreaktoren in eine RBVC-basierte Lehr-/Lernumgebung eingebunden. Der Kurs fand im Rahmen des von der EU geförderten Projektes RAPHAEL [103] statt. Während des Kurses wurde das Simulationsprogramm in einer speziell für diesen Anwendungsfall optimierten Sicht für Übungen der Studierenden eingesetzt. Die Studierenden konnten Simulationsrechnungen zur Reaktivitätsänderung bei schnellen Transienten eigenständig durchführen und durch eigenen Versuch sehen, dass der untersuchte HTR-Modul-Reaktor [104] selbst beim Abschalten der beiden SCRAM⁴⁷-Signale in Kombination mit schnellen Transienten inhärent sicher bleibt.

Bisherige Situation

Das Programm Wkind ist ein Spezialprogramm zur Simulation, das die neutronenphysikalischen Prozesse mit der Thermohydraulik in einem Hochtemperaturreaktor-Kern koppelt und mit dessen Hilfe die Transienten des Reaktors berechnet werden können. Wkind setzt neben der entsprechenden Fachkenntnis über die physikalischen Gegebenheiten auch einen entsprechend hohen Einarbeitungsaufwand voraus, um die vorgegebenen Eingaben in kryptischen Text-Dateien vornehmen zu können. Die Simulationsergebnisse des Programms bestehen ebenfalls aus kryptischen Text-Dateien, die manuell und je nach Anwendungsfall speziell visualisiert werden.

⁴⁷ *Safety Control Rod Axe Man* (SCRAM) ist die Bezeichnung für die Schnellabschaltung eines Kernreaktors.

Mehrwert der Neuerungen

Der Mehrwert der Neuerungen liegt darin, dass das Programm nicht mehr über kryptische Ein- und Ausgabedateien angesprochen wird und somit auch im Bereich der Lehre eingesetzt werden kann. Dies wird durch die rollenabhängigen Sichten ermöglicht. Im Rahmen der Übungen kann jeder Nutzer seine Lerngeschwindigkeit individuell bestimmen. Aber auch Übungen, bei denen in Gruppen auf das System zugegriffen wird, sind möglich. Der Einsatz des Simulationsprogramms in der Lehre bewirkt eine Intensivierung des Lernvorgangs.

Warum Rollen mit gekoppelter Sichtensteuerung?

Die Anwendung eines Simulationsprogramms als Bestandteil einer gemeinsamen Wissensbasis ist nur dann sinnvoll, wenn die Oberflächen und die gesamte Umgebung an die Rolle eines für die Rolle typischen Benutzers angepasst werden.

Rolle	Optimierte Sicht
Dozent	<ul style="list-style-type: none">• Erstellung eines Übungsbeispiels auf Basis einer hinterlegten Geometrie für Studierende.• Festlegung sinnvoller Grenzen für die Eingaben zur Eingabeüberprüfung.• Kontrolle des Nutzungsverhaltens der Studierenden.
Studierender	<ul style="list-style-type: none">• Absolvieren eines Übungsbeispiels.• Visualisierung der Übungsergebnisse.
Fortgeschrittener Studierender	<ul style="list-style-type: none">• Absolvieren erweiterter Übungsbeispiele.• Visualisierung der erweiterten Übungsergebnisse.
Experte	<ul style="list-style-type: none">• Durchführen von Auslegungsrechnungen, Parameterstudien, etc.• Veränderung aller simulationsrelevanten Parameter.• Erstellung neuer Geometrien.• Visualisierung der Berechnungsergebnisse.

Tab. 5: Zuweisung der Rollen zu optimierten Sichten

Für dieses Fallbeispiel sind dies die Rollen Dozent, Studierender, fortgeschrittener Studierender und Experte. Erst durch die Kopplung der Rollen mit der Steuerung der Sichten sind optimierte Ein- und Ausgaben der Oberfläche möglich, in welchen die Berechnungen und auch weiterführende Arbeiten wie beispielsweise Kontrollen

durch den Dozenten durchgeführt werden können. In Tab. 5 sind für das Programm Wkind diese Rollen und die damit gekoppelten, optimierten Sichten aufgeführt.

Welcher Aufwand ist für eine Umsetzung notwendig?

In der technischen Umsetzung wird das ausführbare Simulationsprogramm Wkind durch einen sogenannten Wrapper⁴⁸ umgeben, der eine Steuerung aus einer entsprechenden Komponente ermöglicht. Um eine bessere Lastverteilung realisieren zu können, wird das durch den Wrapper umschlossene Simulationsprogramm auf mehrere Server verteilt. Die Lastverteilung erfolgt unabhängig von der Basisarchitektur. Der Lastverteiler wird über die Dienste-Schnittstelle an die Basisarchitektur angebunden. Der Aufwand für die Umsetzung hängt in diesem Fallbeispiel von zwei Punkten ab.

1. Anzahl der erwarteten, parallelen Zugriffe: Hierbei muss eine Entscheidung anhand der geplanten gleichzeitigen Zugriffe der Benutzer getroffen werden und eine entsprechende Lastverteilung vorgenommen werden.
2. Detaillierungsgrad der Parametrierung: Hierbei ist entscheidend, ob in den Sichten alle Eingabeparameter des Programms benötigt werden, oder ob nur gewisse Einstellungen vorgenommen werden können.

Für die Oberfläche wurde eine spezielle Sicht erstellt. Abb. 33 zeigt die für den Kenntnisstand der Studierenden angepasste Formulareingabe, die nur die Änderung der für die Übung relevanten Simulationsparameter zulässt, die Eingabewerte aber gleichzeitig auf Konsistenz überprüft. Der Aufwand für die Erstellung der Formulareingabe hängt von der Anzahl und der Komplexität der Parameter und den notwendigen Konsistenzprüfungen ab.

⁴⁸ Ein *Wrapper* ist ein Programm, das ein anderes Programm umschließt. Er wird beispielsweise dann eingesetzt, wenn das aufgerufene Programm in einer anderen, inkompatiblen Programmiersprache erstellt wurde. Er bildet die Schnittstelle zwischen dem aufrufenden und dem aufgerufenen Programm.

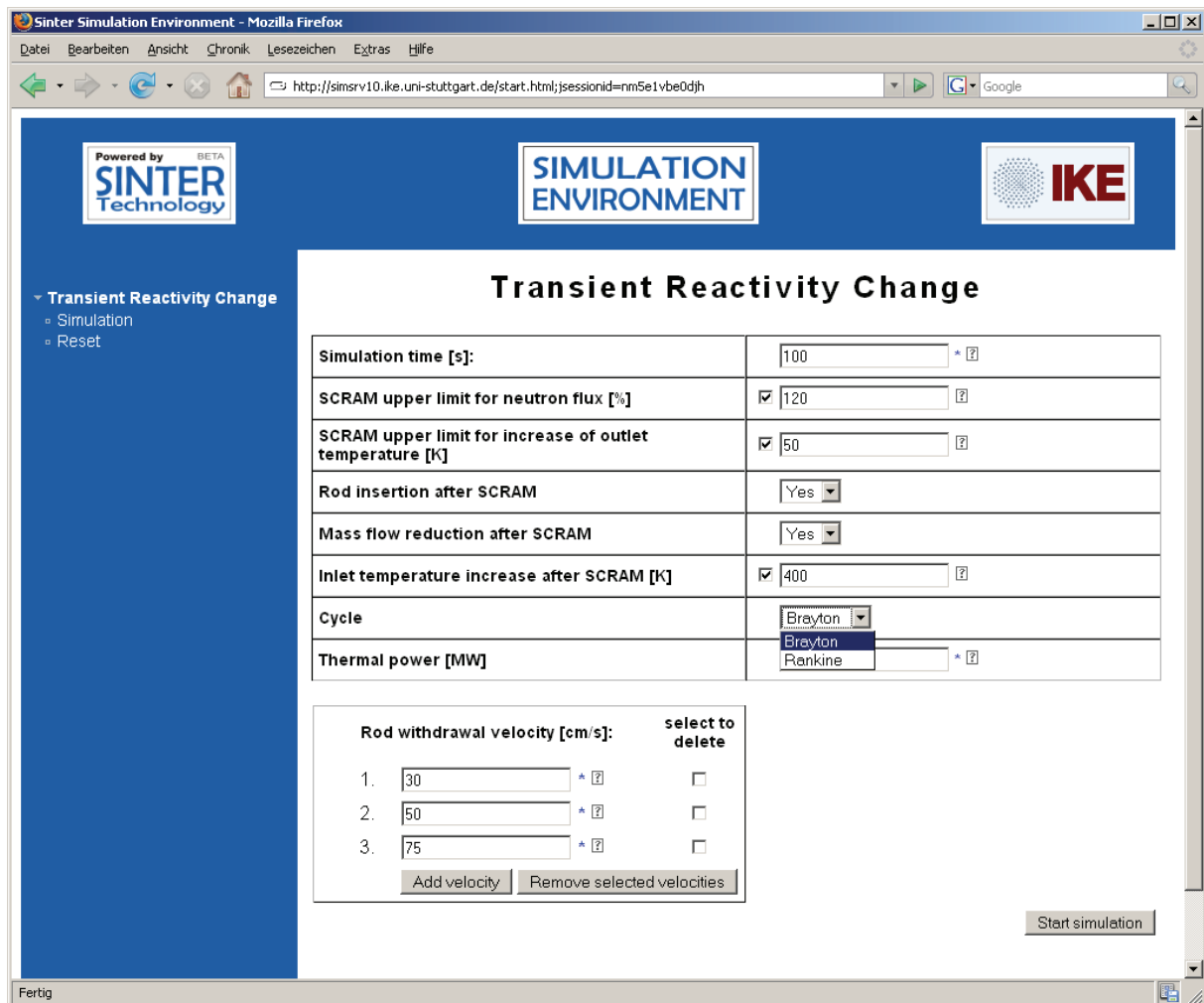


Abb. 33: Sicht für Studierende: Parameterformular des Simulationsprogramms Wkind

Für die Visualisierung wurde ebenfalls eine optimierte Sicht erstellt, in der sämtliche relevanten Ergebnisse graphisch dargestellt werden. Abb. 34 zeigt beispielhaft ein solches Einzelergebnis, das vollkommen automatisch generiert wurde.

Abschließende Bewertung des Fallbeispiels

Das Fallbeispiel Wkind setzt das Szenario Simulationsplattform für Forschung und Lehre aus Kapitel 3.1.1 um und zeigt, dass ein bestehendes Simulationsprogramm, das eigentlich für Experten geschrieben wurde, eine Mehrfachverwendung in der Lehre erfahren kann, wenn dafür Rollen mit optimierten Sichten geschaffen werden.

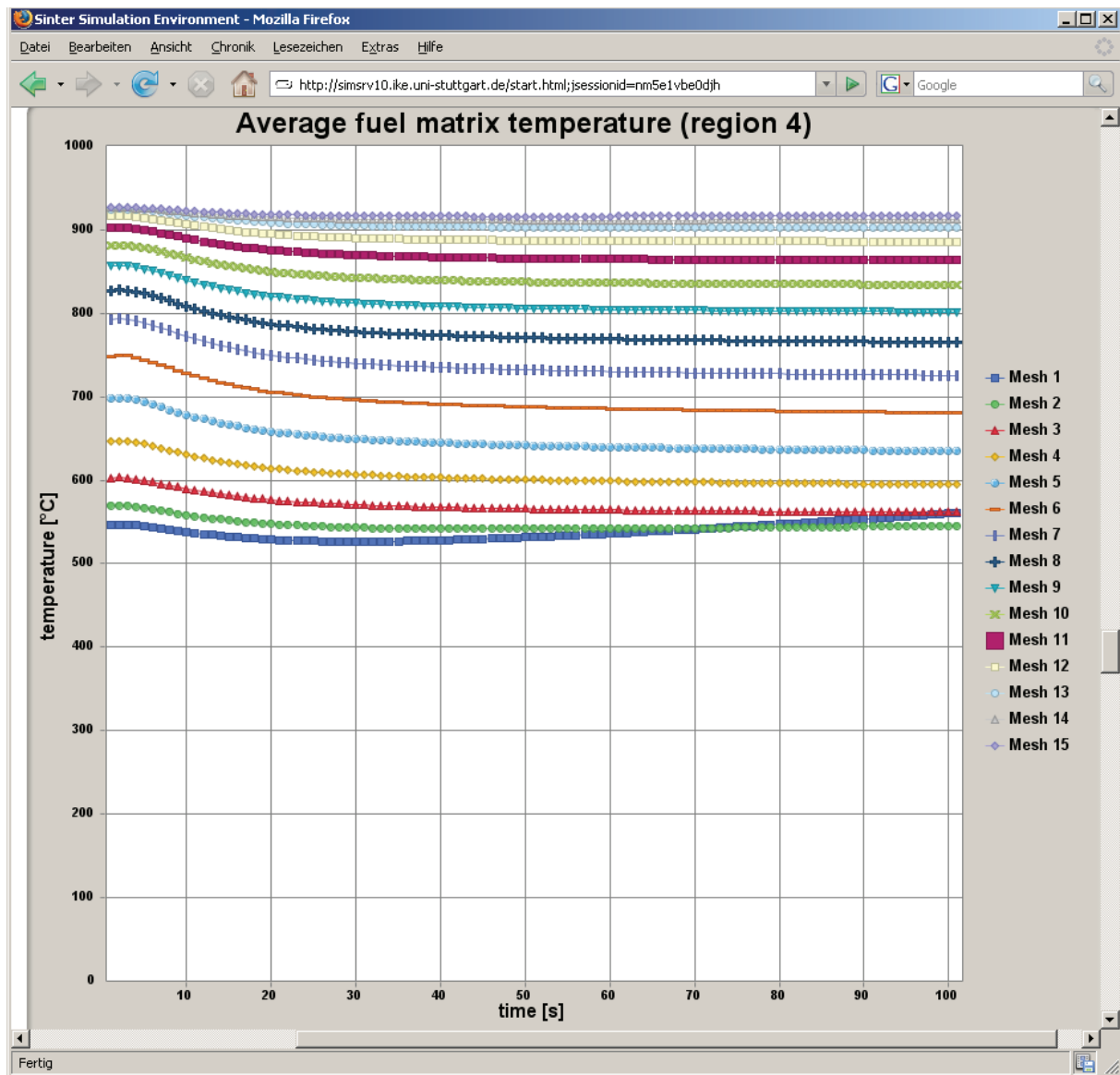


Abb. 34: Sicht für Studierende: Visualisierung eines Teilergebnisses des Simulationsprogramms Wkind

5.2 Fallbeispiel SINTER Network und Überführung in SINTER XT

Bei SINTER XT handelt es sich um das Nachfolgesystem zu SINTER Network, dem Fallbeispiel aus Kapitel 1.1 zum Thema Systeme für gespeichertes und abrufbares Wissen. Es befindet sich momentan im letzten Entwicklungsstadium und setzt das gesamte in dieser Arbeit eingeführte RBVC-Modell um.

Bisherige Situation

Das Informationssystem SINTER Network wurde bisher hauptsächlich dazu verwendet, Forschungsberichte im Rahmen des geschlossenen Benutzernetzwerks in Form einer Online-Bibliothek auszutauschen.

Mehrwert der Neuerungen

Der Mehrwert der Neuerungen liegt darin, dass unterschiedliche Sichten für den Zugriff auf die Inhalte des Informationssystems eingeführt wurden. Da diese Sichten direkt mit den Rollen der Zugriffssteuerung gekoppelt sind, können die eingestellten Forschungsberichte später problemlos in eine gemeinsame Wissensbasis integriert werden.

Warum Rollen mit gekoppelter Sichtensteuerung?

Die direkte Kopplung der Rollen mit der Sichtensteuerung resultiert in diesem Fallbeispiel in vielen neuen Anwendungsmöglichkeiten der eingestellten Forschungsberichte. Diese können nun problemlos in anderen Kontexten und von anderen berechtigten Interessengruppen (z.B. Regulierungsbehörden, Herstellerbetrieben, Energieversorgungsunternehmen und Hochschulen) verwendet werden und so einen Teil einer gemeinsamen Wissensbasis bilden. Bisher hätte der Zugriff auf diese grobgranularen Gesamtberichte meist verwehrt werden müssen, da nicht alle Informationen eines Berichts allen Benutzern zur Verfügung gestellt werden dürfen. Die neuen Filtermöglichkeiten, die durch die Sichtensteuerung umgesetzt werden, erlauben beispielsweise, interne Berichtsteile auszufiltern, da diese ausschließlich speziell privilegierten Rollen vorbehalten sind.

Welcher Aufwand ist für eine Umsetzung notwendig?

Der Aufwand für die Umsetzung des Systems SINTER XT im Bereich der Berichteingabe beschränkt sich darauf, die Sichten für einzelne Rollen und damit verbunden einzelne Berichtsarten zu generieren. Für neue Berichtsarten müssen die jeweiligen Verwaltungssichten angepasst und ergänzt werden. Für die Abfrage der eingestellten Berichte müssen je nach Privilegien spezielle Sichten erstellt werden.

Abschließende Bewertung des Fallbeispiels

Das Fallbeispiel SINTER XT führt den ersten Ansatz seines Vorgängers zum Aufbau eines einheitlichen Informationssystems mit verschiedenen Untersystemen im Bereich Kerntechnik auf Grundlage der im Rahmen dieser Arbeit entwickelten Basisarchitektur konsequent fort. Neben den Funktionen des Vorgängers SINTER Network als Kooperationswerkzeug wird das neue SINTER XT in seiner späteren Ausbaustu-

fe direkt an die SINTER Simulationsplattform angebunden werden, welche als Lehr-/Lernumgebung für Kurse beispielsweise im Rahmen der ENEN⁴⁹-Association [105] und des EU-Projekts RAPHAEL⁵⁰ [103] (vgl. 5.1) dient. Durch die Web-Service-Schnittstelle (vgl. 4.7.6) ist eine Integration der verteilt vorliegenden Daten anderer Hochschulen, Forschungsorganisationen und Herstellerbetrieben möglich. Weiter ist geplant, auch Informationen, die im Rahmen des Systems NEPTUNO CS (vgl. 5.4) gesammelt wurden, über diese Schnittstelle bereitzustellen. SINTER XT stellt somit die Grundlagen für eine zukünftige, gemeinsame Wissensbasis im Bereich Kerntechnik zur Verfügung und setzt das Szenario Kombinationssystem – gemeinsame kerntechnische Wissensbasis aus Kapitel 3.1.5 um.

5.3 Fallbeispiel ABR-KFUE und Überführung in ABR-Research

Der Berechnungskern des Programms ABR-KFUE, Fallbeispiel für ein Trainingssystem aus Kapitel 1.6.3, wurde im Rahmen der beiden Projekte AJA [106] und KEWA [107] [108] des Umweltministeriums Baden-Württemberg in ein RBVC-basiertes Forschungs- und Trainingssystem namens ABR-Research eingebunden. Das System ABR-Research setzt die beiden Szenarien Simulationsplattform für Forschung und Lehre (Kapitel 3.1.1) und Integrationssystem (Kapitel 3.1.2) um.

Bisherige Situation

ABR-KFUE konnte in seiner bisherigen Form ausschließlich für Übungen zur Bedienung der Oberfläche des Klienten eingesetzt werden, da es sich um ein reines Notfallschutzsystem handelt.

Mehrwert der Neuerungen

In ABR-Research können erstmals Berechnungen des ABR-KFUE-Systems im Bereich der Forschung und Lehre eingesetzt werden. Das neue System erlaubt eine Mehrfachverwendung des Berechnungskerns für spezielle Forschungs- und Trainingsaufgaben, die mit den Einschränkungen eines im Betrieb befindlichen Notfallschutzsystems nicht durchgeführt werden können.

⁴⁹ European Nuclear Education Network

⁵⁰ ReActor for Process heat, Hydrogen And Electricity generation

Warum Rollen mit gekoppelter Sichtensteuerung?

Vom Umweltministerium Baden-Württemberg wird ABR-Research in einer speziellen Sicht dafür eingesetzt, Szenarien für Übungen des Katastrophenschutzes vorzubereiten. Eine weitere Sicht ermöglicht dem Experten das Testen von Funktionserweiterungen des Berechnungskerns, beispielsweise Prognoserechnungen unter Einbeziehung der Vorhersagedaten des Deutschen Wetterdienstes. Bei erfolgreichem Verlauf der Tests können die Erweiterungen in das Notfallschutzsystem ABR-KFUE übernommen werden.

Welcher Aufwand ist für eine Umsetzung notwendig?

In der technischen Umsetzung wurde eine Workflow-Engine in einer Komponente angebunden, die den Ablauf der ausführbaren Simulationsprogramme des Berechnungskerns steuert. Die Formulare der Oberfläche wurden mit GWT umgesetzt, welche in jeder Sicht eine flüssige Eingabe erlauben. Für die Visualisierung der Ergebnisse wurde die Google Maps API [109] [110] verwendet, auf der Ausbreitungsfahnen und andere Ergebnisse in frei skalierbaren Karten-, Satellitenbild- und Hybriddarstellungen als Overlay-Objekte abgebildet werden können.

Abschließende Bewertung des Fallbeispiels

Das Fallbeispiel ABR-Research zeigt, welcher Mehrwert aus der Mehrfachverwendung einer bestehenden Anwendung entstehen kann, wenn diese mit rollenbasierten Sichten kombiniert und in einer Integrationsumgebung ausgeführt werden. Neben der Vorbereitung von Übungen des Katastrophenschutzes lässt sich das System auch für Forschungsaufgaben und zum Test neuer Funktionen einsetzen. Zukünftig ist angedacht, dass in einer weiteren Sicht die Ausbreitung der Schadstoffe von mobilen Emissionsstandorten, beispielsweise für Castor⁵¹-Transporte, berechnet werden kann.

5.4 NEPTUNO-CS

Das System NEPTUNO⁵²-CS [111] ist ein kerntechnisches Informationssystem des FP-6, in dem Informationen über Kurse, Vorlesungen, Seminare und Trainingseinheiten von über 40 europäischen Hochschulen und Forschungsorganisationen integriert

⁵¹ Cask for Storage and Transport of Radioactive Material

⁵² Nuclear European Platform for Training and UNiversity Organisations

wurden. Das System bietet die Möglichkeit, Veranstaltungen europaweit anzukündigen. Eine Datenbank mit Informationen über kerntechnische Einrichtungen wurde ebenfalls integriert. Das System NEPTUNO-CS ist eine Kombination der beiden Szenarien Integrationssystem (Kapitel 3.1.2) und kerntechnisches Informationssystem (Kapitel 3.1.4).

Bisherige Situation

Informationen über Kurse, Vorlesung, Seminare und Trainingseinheiten und Einrichtungen im Fachbereich Kerntechnik sind bislang nicht über eine zentrale Stelle im Internet aufrufbar gewesen. Jede Interessengruppe veröffentlichte die angebotenen Kurse auf ihrer eigenen Präsenz im Internet.

Mehrwert der Neuerungen

Mit NEPTUNO-CS wurde erstmalig ein Informationssystem geschaffen, in dem zentral alle oben genannten Daten eingetragen und im europäischen Rahmen veröffentlicht werden können. Die gesammelten Daten aus der Kursdatenbank sind, wenn sie nicht ständig gepflegt werden, nur über einen kurzen Zeitraum aktuell.

The screenshot shows the NEPTUNO CS website interface. The main heading is "Courses not approved by the organising institution". Below this, there is explanatory text and instructions for users. A table lists two courses, both titled "Accelerators", offered by "Ecole Nationale Supérieure des Techniques Industrielles et des Mines de Nantes". Each course entry has buttons for "Report Pdf", "Edit", and "Take over", along with a "Select to remove" checkbox.

Nr.	Course Title	Course Type	Organising Institution	Report Pdf	Edit	Take over	Select to remove
1	Accelerators	education	ECOLE NATIONALE SUPERIEURE DES TECHNIQUES INDUSTRIELLES ET DES MINES DE NANTES La Chantrerie, 4 rue A. Kastler, BP 20722, F44307 Nantes Cedex 3	Report Pdf	Edit	Take over	<input type="checkbox"/>
2	Accelerators	education	Ecole Nationale Supérieure des Techniques Industrielles et des Mines de Nantes La Chantrerie, 4	Report Pdf	Edit	Take over	<input type="checkbox"/>

Abb. 35: Übernahme von bisher nicht gepflegten Kursdaten durch eine Organisation

Eine zentrale Aktualisierung ist sehr aufwendig, deswegen wurde in NEPTUNO-CS ein Ansatz gewählt (Abb. 35), der es den Angehörigen einer Organisation erlaubt, bereits angelegte, aber nicht gepflegte Kurse zu ihren eigenen Kursen zu übernehmen. So können angemeldete Organisationen selbständig Aktualisierungen ihrer Kurse vornehmen.

Warum Rollen mit gekoppelter Sichtensteuerung?

Die Anwendung der entwickelten Basisarchitektur und des zugrundeliegenden RBVC-Modells unterstützt die Umsetzung der verteilten Administration optimal. Durch die Steuerung der Sichten können Benutzer je nach Rolle in speziellen Eingabefeldern die notwendigen Änderungen vornehmen.

Welcher Aufwand ist für eine Umsetzung notwendig?

Bei der Umsetzung unter Anwendung des RBVC-Modells müssen Rollen und Sichten für Organisationsmitglieder, Organisationsleiter und für Studierende mit optimierten Sichten erstellt werden.

Abschließende Bewertung des Fallbeispiels

Das Fallbeispiel NEPTUNO-CS zeigt, dass die verteilte Administration des RBVC-Modells sehr leicht umgesetzt und delegiert werden kann. Die Sichten vereinfachen die Pflege und Aktualisierung der eingestellten Daten.

5.5 GRS-FBW

Das System GRS-FBW [112] ist ein Dokumentenverwaltungssystem, in dem die Fortschrittsberichte eingestellt werden, welche vom Bundesministerium für Wirtschaft und Technologie gefördert und von der *Gesellschaft für Anlagen- und Reaktorsicherheit mbH* (GRS) betreut werden. Die Berichte werden direkt von den Forschungsstellen eingestellt und stammen aus dem Bereich der Reaktorsicherheit. Nach der Veröffentlichung können Berichte nach diversen Kriterien gesucht werden (Abb. 36). Bis zur ihrer Veröffentlichung werden die Berichte von der Forschungsbetreuung der GRS betreut und verwaltet.

GRS-FBW setzt das Szenario kerntechnisches Informationssystem aus Kapitel 3.1.4 um.



Abb. 36: Auswahl der Kriterien zur Suche nach veröffentlichten Fortschrittsberichten in GRS-FBW

Bisherige Situation

Die Fortschrittsberichte wurden bisher als Einzelberichte in SINTER Network (Kap. 5.2) eingestellt. Die Berichte eines Berichtszeitraums wurden von der Abteilung Forschungsberichtswesen der GRS jeweils manuell zu einer Druckfassung im Rahmen eines Gesamtberichts zusammengestellt. Bei der Zusammenstellung traten häufig Konsistenzprobleme auf, wie beispielsweise fehlerhafte Kopf- und Fußzeilen, inkonsistente Kapitelüberschriften, falsche Schriftarten, etc.

Mehrwert der Neuerungen

Die Einführung des GRS-FBW-Systems vereinfacht maßgeblich die Eingabe der Fortschrittsberichte. Es wird eine Unterscheidung zwischen externen (öffentlichen) und internen (nicht öffentlichen) Berichtsteilen eingeführt. Die Einzelberichte können von den jeweiligen Forschungsstellen direkt in GRS-FBW erstellt werden. Nachdem alle Berichte eines Berichtszeitraums erfasst sind, können diese automatisch in die Druckfassung in Form eines Gesamtberichts umgewandelt und der Druckerei übergeben werden. Dadurch können die früher existierenden Konsistenzprobleme, wel-

che durch die manuelle Zusammenstellung des Gesamtberichts entstanden, beseitigt werden.

Warum Rollen mit gekoppelter Sichtensteuerung?

Dieses Beispiel veranschaulicht, wie durch individuell abgestimmte Menüs, die sich aus den jeweiligen Rollen und den dazugehörigen Sichten eines Benutzers ergeben, Oberflächen entscheidend vereinfacht werden können. Dies gilt insbesondere für die Fälle, in denen ein Benutzer mehrere Rollen gleichzeitig im System erfüllt (Beispiel: Rollen Qualitätskontrolle und Forschungsbetreuung gleichzeitig). Hierbei werden die Oberflächen an die jeweiligen Aufgaben optimiert und in einer gemeinsamen Oberfläche dargestellt.

Welcher Aufwand ist für eine Umsetzung notwendig?

Die Anwendung des RBVC-Modells vereinfacht spürbar die Entwicklung des GRS-FBW-Systems. Neben den erhöhten Anforderungen an die Zugriffsberechtigungen werden unterschiedliche Sichten für die aufgabenbedingten Rollen des Systems generiert:

- Autoren an den Forschungsstellen
- Qualitätskontrolle
- Forschungsbetreuung
- Zentraladministration der Forschungsbetreuung

Das Menü des Benutzers wird abhängig von dessen Rollen angepasst, so dass jeweils nur die dort verfügbaren Funktionsausprägungen erscheinen.

Abschließende Bewertung des Fallbeispiels

Die Einführung des GRS-FBW-Systems zeigt, wie durch eine rollenabhängige Sichtensteuerung gerade in Bereichen, in denen die Benutzer mehrere Rollen haben, maßgebliche Vereinfachungen bei der Erstellung und Verwaltung der Fortschrittsberichte erreicht werden können. Die Berichte, die in GRS-FBW erfasst sind lassen sich später problemlos in eine gemeinsame Wissensbasis integrieren, ohne dass diese neu erfasst werden müssen. Dabei können die internen Berichtsteile über entsprechende Sichten herausgefiltert werden.

6 Zusammenfassung

Die Motivation zur Erstellung dieser Arbeit lag im Fehlen eines allgemeinen Benutzermodells mit Rollen, Rechten und Sichten für eine gemeinsame Wissensbasis im Bereich Kerntechnik. Ein solches Modell ermöglicht es, strukturiertes Wissen zu integrieren und effizient zu verwalten. Das Wissen kann dabei über mehrere Forschungsorganisationen, Hochschulen und Herstellerbetriebe verteilt sein. Es muss ein Rollenmodell mit Integrations- und Verteilungsaspekten vereinen. Dabei sind die sehr hohen Sicherheitsanforderungen des Fachbereichs zu beachten, da durch eine Freigabe des Zugriffs über das Internet die jeweils organisationseigene Netzwerksicherheit nicht gefährdet werden darf. Um das Modell auch im Bereich des elektronischen Lernens einsetzen zu können, muss eine Integration auch über unterschiedliche Quellen wie Simulationsprogramme, Informationssysteme und Lerneinheiten möglich sein.

In der vorliegenden Arbeit wurde ein allgemeines Rollenmodell für den Bereich Kerntechnik vorgestellt, das neben der reinen Zugriffssteuerung eine direkte Kopplung der Rollen mit optimierten Sichten ermöglicht. Das daraus entstandene RBVC-Modell wurde aus dem theoretischen Rollenmodell RBAC abgeleitet, das in einer ANSI-Norm vorliegt. Ausgehend von fünf typischen Szenarien für den Fachbereich wurden die Rahmenbedingungen für das Rollenmodell definiert. Diese wurden umgesetzt und das RBAC-Modell hierfür entsprechend erweitert.

Die Erweiterungen erlauben eine Integration des Wissens, das aus unterschiedlichen Bereichen stammen kann, beispielsweise aus Simulationsprogrammen, Datenbanken, Wissensbanken, Informationssystemen, Lehr-/Lernsystemen und Trainingssystemen. Dazu wurden Ressourcentypen eingeführt, die sicherstellen, dass die jeweiligen Operationen nur auf Objekte in geeigneter Paarung angewandt werden können. Das RBVC-Modell ermöglicht den direkten Einfluss auf das Verhalten der Rollenhierarchien. Das RBAC-Modell sieht hierzu zwei optionale Hierarchiearten vor, die exklusiv verwendet werden müssen und nicht kombiniert werden können. Dies reicht in der Praxis bei komplexen Systemen nicht aus. Deshalb wurde das RBVC-Modell um Rollentypen erweitert, die eine parallele, gleichzeitige Verwendung unterschiedlicher Rollenhierarchien erlauben. Das Modell führt die Möglichkeit einer verteilten Administration ein, die aus eigenständigen Verwaltungseinheiten, beispielweise aus ein-

zelen Forschungsstellen unterschiedlicher Organisationen, bestehen können. Neu ist auch die direkte Kopplung der Zugriffssteuerung mit rollenbasierten Sichten. Die Rollen, die ein Benutzer in einem System einnimmt, ergeben sich aus seinen Aufgaben. Diese setzen einen entsprechenden Kenntnisstand voraus. Die Sichten können je nach Anwendungsfall, der sich aus der Aufgabe eines typischen Benutzers ergibt, unter Berücksichtigung des dafür benötigten Kenntnisstandes optimiert werden. Eine Sicht repräsentiert somit eine Funktion in einer speziell aufbereiteten Ausprägung.

Aufbauend auf das entwickelte RBVC-Modell wurde das Konzept einer Basisarchitektur für web-basierte, verteilte und integrierende Systeme entwickelt, welche auch die Möglichkeit zur Fernautorisierung in anderen Systemen anbietet. Sie besteht aus einer Dreischichtarchitektur, die in eine Präsentations-, Logik-, und Datenschicht aufgeteilt ist. Diese Architektur ermöglicht im Gegensatz zur MVC-Architektur eine vergleichsweise einfache Verteilung über Web-Service-Schnittstellen, da die Präsentationsschicht streng von der Datenschicht getrennt und sämtliche Kommunikation zwischen den beiden Schichten durch die zentrale Logikschicht entkoppelt wird.

Die Präsentationsschicht greift auf die zentrale Logikschicht über das Internet zu und besteht im Regelfall aus einem Browser. Ergänzend ist auch der Zugriff über einen Web-Service an dieser Stelle angedacht. Unabhängig von der Zugriffsart gehen Anfragen immer bei der Zugriffssteuerung der Logikschicht ein, in der die rollenbasierte Autorisierung stattfindet. Nachdem eine berechtigte Anfrage an die jeweilige Anwendung weitergegeben und abgearbeitet wurde, übergibt diese die Antwort an die Sichtensteuerung, welche die Antwort von der Sichtengenerierung rollenentsprechend aufbereiten lässt. Sobald dies geschehen ist, wird die Antwort von der Sichtensteuerung an die Präsentationsschicht zurückgegeben.

In der Logikschicht werden die Anwendungen des Systems über Komponentenschnittstellen angebunden. Die Anwendungen enthalten im Kontext der Zugriffssteuerung die entsprechend ausführbaren Operationen und können lokal oder aber auch über einen Web-Service an die Logikschicht angebunden werden. Sie greifen auf die anwendungsrelevanten Objekte der Datenschicht zu. Die Quellen dieser Schicht können ebenfalls lokal oder über eine Web-Service-Schnittstelle angebunden sein.

Die Implementierung der Logikschicht basiert auf dem Komponentenframework Apache Cocoon, das durch Umsetzung von *Separation of Concerns* in klar voneinander

trennbare funktionelle Bereiche aufgeteilt ist. Die zugrundeliegenden Komponenten sind unter Berücksichtigung des Entwurfsmusters *Inversion of Control* realisiert mit dem Vorteil, dass sie nur lose miteinander gekoppelt werden und deswegen einfach gegen andere Komponenten ausgetauscht werden können. Die Präsentationsschicht besteht aus einem dünnen Klienten. Sie kann allerdings optional mit AJAX-Technologien erweitert werden, welche einen Teil der Formular-Steuerungsfunktionen in den Browser und somit in die Präsentationsschicht verlegen. Dies gepaart mit der asynchronen Serverkommunikation erlaubt den Aufbau von Oberflächen, die für den Benutzer in der Bedienung wesentlich flüssiger wirken als herkömmliche Webanwendungen. In der Datenschicht können beliebige Datenquellen angebunden werden, sofern für den Zugriff eine entsprechende Komponente zur Verfügung steht. Die Anbindung relationaler Datenbanken erfolgt über das Persistenz-Framework Hibernate.

Das RBVC-Modell und das Konzept der Basisarchitektur wurden in diversen Softwareprojekten angewandt und umgesetzt und haben ihre Tragfähigkeit für Systeme im Bereich Kerntechnik bewiesen. Es hat sich gezeigt, dass die Einführung rollenbasierter Sichten ein wirkungsvolles Mittel sind, Funktionen in einer optimierten Ausprägung den Aufgaben typischer Benutzer anzupassen. Die Integrations- und Verteilungsfähigkeit ergibt viele neue Kombinationsmöglichkeiten, in denen Wissen zusammengeführt werden kann. Der Vorteil dieser Möglichkeiten eröffnet insbesondere für die Lehre in der Kerntechnik eine neue Dimension. Hier können neue Übungsformen in Lehr-/Lernsystemen geschaffen werden, indem eine Integration von Simulationsprogrammen in vereinfachenden Sichten stattfindet. Sie erhöhen den Memorierungsgrad einer Lehrveranstaltung und gewährleisten somit deren Nachhaltigkeit. Auch im Bereich der Forschung eröffnet der konsistente Zugriff über ein organisationsübergreifendes, gemeinsames Rollenmodell neue Möglichkeiten auf verteilte Wissensquellen, deren Freigabe bis dato an unzureichenden Zugriffssteuerungsmethoden scheiterten, zuzugreifen.

7 Ausblick

Im Rahmen der vorliegenden Arbeit wurde das RBVC-Modell für den Bereich Kerntechnik entwickelt. Es erweitert das RBAC-Referenzmodell, verbessert die Integrationsfähigkeiten unterschiedlicher Quellobjekte und erlaubt eine verteilte Administration. Weiter verbindet es die reine Zugriffssteuerung mit der Steuerung rollenbasierter Sichten. Die darauf aufbauende Basisarchitektur unterstützt Integrations- und Verteilungsmöglichkeiten und bietet neue Kombinationsmöglichkeiten der beschriebenen Wissensquellen. In weiteren Untersuchungen ist zu klären, welche neuen Wissensquellen für den Bereich Kerntechnik erschlossen werden können.

Im Bereich der Lehr-/Lernsysteme und Simulationssysteme ist zu untersuchen, ob ein vom RBVC-Modell abgeleitetes Modell um kooperative Methoden erweitert werden kann. Hierbei muss geprüft werden, ob die Rollen der Benutzer als Kriterium geeignet sind, um aufgrund der Rollendefinition eine Suche nach Personen mit spezifischem Wissen, z.B. Experten eines Fachgebiets, durchführen zu können. Erste Untersuchungen hierzu wurden bereits in [113] durchgeführt.

Das RBVC-Modell kann leicht auf andere Bereiche in den Ingenieurwissenschaften übertragen werden. Dabei ist zu untersuchen, in welchen Kontexten der Einsatz sinnvoll ist und ob hierzu Erweiterungen benötigt werden.

Diese Übertragbarkeit kombiniert mit der Verbindung der reinen Zugriffssteuerung und der Steuerung rollenbasierter Sichten kann das Systemverhalten, die Datensicherheit und die Ergonomie der Benutzeroberflächen auch außerhalb des Bereichs der Ingenieurwissenschaften verbessern. Hierbei ist der Einsatz des RBVC-Modells im Bereich des e-Governments der öffentlichen Verwaltung in Baden-Württemberg denkbar. Dabei muss in einem ersten Schritt die Frage geklärt werden, für welche Anwendungen eine direkte Integration über Web-Services und unter Verwendung des RBVC-Modells in Verwaltungsportale sinnvoll ist. Untersuchungen hierzu können beispielsweise im Rahmen des vom Umweltministerium Baden-Württemberg geförderten Forschungs- und Entwicklungsvorhabens KEWA⁵³ durchgeführt werden.

⁵³ Kooperative Entwicklung wirtschaftlicher Anwendungen für Umwelt und Verkehr in neuen Verwaltungsstrukturen

Literaturverzeichnis

- [1] Fernandez-Ruiz, P.; Forsström, H.; Van Goethem, G.: *The sixth Euratom framework programme 2003-2006: a driving force for the construction of the Nuclear European Research Area*. In: Nuclear Engineering and Design, Vol. 235, Iss. 2-4, p. 127-137, Elsevier, Lausanne 2005.
- [2] *Vertrag von Amsterdam zur Änderung des Vertrags über die Europäische Union, der Verträge zur Gründung der Europäischen Gemeinschaften sowie einiger damit zusammenhängender Rechtsakte*. In: Amtsblatt C 340, 1997, <http://europa.eu.int/eur-lex/de/treaties/dat/amsterdam.html>,
gesehen am 19.05.2007.
- [3] Community Research and Development Information Service (Hg): *EURATOM: Research and training in the field of Nuclear Energy*.
<http://cordis.europa.eu/fp5-euratom/home.html>,
gesehen am 16.05.2007.
- [4] Community Research and Development Information Service (Hg): *Sixth Framework Programme*.
<http://cordis.europa.eu/fp6>,
gesehen am 17.05.2007.
- [5] Van Goethem, G.: *Towards a common knowledge base for nuclear research: a challenge for the stakeholders community and for the EC*. International Conference on Nuclear Knowledge Management, Saclay, 2004,
<http://www.iaea.org/km/cnkm/papers/ecgoethem.pdf>,
gesehen am 17.05.2007.
- [6] Community Research and Development Information Service (Hg): *European Research Area*.
<http://cordis.europa.eu/era/>,
gesehen am 18.05.2007.
- [7] Community Research and Development Information Service (Hg): *Integrated Projects (IPs)*.

- http://cordis.europa.eu/fp6/instr_ip.htm,
gesehen am 18.05.2007.
- [8] Community Research and Development Information Service (Hg): *Networks of Excellence (NoE)*.
http://cordis.europa.eu/fp6/instr_noe.htm,
gesehen am 18.05.2007.
- [9] *Beschluss Nr. 1639/2006/EG des Europäischen Parlaments und des Rates vom 24. Oktober 2006 zur Einrichtung eines Rahmenprogramms für Wettbewerbsfähigkeit und Innovation (2007-2013)*. In: Amtsblatt der Europäischen Union, Luxemburg, 2006.
http://eur-lex.europa.eu/LexUriServ/site/de/oj/2006/l_310/l_31020061109de00150040.pdf,
gesehen am 18.05.2007.
- [10] Moos, F. et al.: *European Master of Science in Nuclear Engineering*. In: Nuclear Engineering and Design, Vol. 235, Iss. 2-4, p. 165-172, Elsevier, Lausanne, 2005.
- [11] Fishwick, P. A.: *Simulation model design and execution: building digital worlds*. Prentice-Hall, Englewood Cliffs, 1995.
- [12] Walter, A.; Schulz, A.; Lohnert, G.: *Comparison of two models for a pebble bed modular reactor core coupled to a Brayton cycle*. In: Nuclear Engineering and Design, Vol. 236, Iss. 5-6, p. 603-614, Elsevier, Lausanne, 2006.
- [13] *SINTER Network – The Sustainable & Innovative Nuclear Technology Evolution – R & D Network*.
http://w2ksrvx.ike.uni-stuttgart.de/sinter_neu/,
gesehen am 20.05.2007.

- [14] Weigele, M.; Achenbach, J.; Piater, A.; Schmidt, F.; Schulz, A.; Sucic, D.: *KFÜ-ABR - Entwicklung eines ABR-Research Systems*. In: Projekt AJA: Anwendung JAVA-basierter und anderer leistungsfähiger Lösungen in den Bereichen Umwelt, Verkehr und Verwaltung, Phase IV – 2003, Mayer-Föll, R.; Keitel, A.; Geiger, W. (Hg), Wissenschaftliche Berichte FZKA 6950, Ministerium für Umwelt und Verkehr Baden-Württemberg, Landesanstalt für Umweltschutz Baden-Württemberg, Forschungszentrum Karlsruhe, Karlsruhe, 2003.
- [15] United States Government Department of Defense (Hg): *Trusted Computer System Evaluation Criteria*. Department of Defense Standard, CSC-STD-001-83, USA, 1983.
- [16] United States Government Department of Defense (Hg): *Trusted Computer System Evaluation Criteria*. Department of Defense Standard, DoD 5200.28-STD, USA, 1985.
- [17] Bundesamt für Sicherheit in der Informationstechnik (Hg): *Kriterien für die Bewertung der Sicherheit von Systemen der Informationstechnik: (ITSEC)*. Bundesanzeiger, Köln, 1992.
- [18] Zentralstelle für Sicherheit in der Informationstechnik (ZSI) (Hg): *IT-Sicherheitskriterien, Kriterien für die Bewertung der Sicherheit von Systemen der Informationstechnik*, Bundesanzeiger, Köln, 1989.
- [19] Common Criteria (Hg): *Common Methodology for Information Technology Security Evaluation: Evaluation methodology*. CCMB-2005-08-004, Version 2.3, 2005, <http://www.commoncriteriaportal.org/public/files/cemv2.3.pdf>, gesehen am 26.01.2007.
- [20] International Organization for Standardization (Hg): *International Standard: ISO/IEC 15408-1: Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model*. 2. Aufl., ISO/IEC 15408-1/2005(E), Genf, 2005.

- [21] International Organization for Standardization (Hg): *International Standard: ISO/IEC 15408-2: Information technology – Security techniques – Evaluation criteria for IT security – Part 2: Security functional requirements*. 2. Aufl., ISO/IEC 15408-2/2005(E), Genf, 2005.
- [22] International Organization for Standardization (Hg): *International Standard: ISO/IEC 15408-3: Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance requirements*. 2. Aufl., ISO/IEC 15408-3/2005(E), Genf, 2005.
- [23] Common Criteria (Hg): *Common Methodology for Information Technology Security Evaluation: Evaluation methodology*. CCMB-2006-09-004, Version 3.1, Rev. 1, 2006,
<http://www.commoncriteriaportal.org/public/files/CEMV3.1R1.pdf>,
gesehen am 26.01.2007.
- [24] Bundesamt für Sicherheit in der Informationstechnik (Hg):
IT-Sicherheitskriterien: Gemeinsame Kriterien für die Prüfung und Bewertung der Sicherheit von Informationstechnik.
<http://www.bsi.bund.de/cc/>,
gesehen am 26.01.2007.
- [25] Olsen, I.; Abrams, M.; *Computer Access Control Policy Choices*. In: *Computers & Security*, Vol. 9 (1990), p. 699-714, Elsevier Advanced Technology Publications, Oxford, 1991.
- [26] Ferraiolo, D.; Kuhn, R.: *Role-Based Access Control*. In: *Proceedings of the NIST-NSA 15th National Computer Security Conference*, p. 554-563, USA, 1992.
- [27] Ferraiolo, D.; Kuhn, R.; Chandramouli, R.: *Role-Based Access Control*. Artech House, Boston, London, 2003.
- [28] American National Standards Institute (Hg): *American National Standard for Information Technology – Role Based Access Control*. ANSI INCITS 359-2004, USA, 2004.

- [29] Bronstein, I.; Semendjajew, K.; Musiol, G.: *Taschenbuch der Mathematik*. 6. Aufl., Harri Deutsch Verlag, Frankfurt am Main, 2005.
- [30] Kuhn, R.: *Mutual exclusion of roles as a means of implementing separation of duty in role-based access control systems*. In: Proceedings of the second ACM workshop on Role-based access control, p. 23-30, ACM Press, New York, 1997.
- [31] Simon, T.; Zurko, M.: *Separation of Duty in Role-Based Environments*. In: Proceedings of 10th IEEE Computer Security Foundations Workshop, p. 183-194, Rockport, 1997.
- [32] Beck, K.: *Extreme Programming*. Addison-Wesley, München, 2003.
- [33] Martin, R.: *Agile Software Development. Principles, Patterns, and Practices*. Prentice Hall International, 2003.
- [34] Bertino, E.: *RBAC models – concepts and trends*. In: Computers & Security, Vol. 22, Iss. 6 (2003), p. 511-514, Elsevier Advanced Technology Publications, Oxford, 2003.
- [35] Sandhu, R.: *Rationale for the RBAC96 family of access control models*. In: Proceedings of the first ACM Workshop on Role-based access control, Art. 9, ACM Press, New York, 1996.
- [36] Sandhu, R.; Coyne, E.; Feinstein, H.; Youman, E.: *Role-Based Access Control Models*. In: IEEE Computer, Vol. 29, No. 2, 1996, p. 38-47, IEEE Computer Society Press, Los Alamitos, 1996.
- [37] Ferraiolo, D.; Cugini, J.; Kuhn, D.: *Role-Based Access Control (RBAC): Features and Motivations*. In: Proceedings of the Annual Computer Security Conference, p. 23-30, ACM Press, New York, 1997.
- [38] Sandhu, R.; Bhamidipati, V.; Munawer, Q.: *The ARBAC97 Model for Role-Based Administration of Roles*. In: ACM Transactions on Information and System Security (TISSEC), Vol. 2, Iss. 1, p. 105-135, ACM Press, New York, 1999.

- [39] Oh, S.; Sandhu, R.: *A Model for Role Administration Using Organisation Structure*. In: Proceedings of the seventh ACM symposium on Access control models and technologies, p. 155-162, ACM Press, New York, 2002.
- [40] Bösing, S.: *Authentifizierung und Autorisierung im elektronischen Rechtsverkehr*. 1. Aufl., Nomos Verlag, Baden-Baden, 2005.
- [41] Bundesamt für Sicherheit in der Informationstechnik (Hg): *Leitfaden IT-Sicherheit*.
<http://www.bsi.de/gshb/Leitfaden/GS-Leitfaden.pdf>,
gesehen am 26.01.2007.
- [42] Spafford, E.: *OPUS: Preventing Weak Password Choices*. Purdue Technical Report CSD-TR 92-028, 1991,
<http://ftp.cerias.purdue.edu/pub/papers/gene-spafford/spaf-OPUS.pdf>,
gesehen am 26.01.2007.
- [43] Bundesamt für Sicherheit in der Informationstechnik (Hg): *Studie: „Evaluierung biometrischer Systeme Fingerabdrucktechnologien – BioFinger“, Öffentlicher Abschlussbericht*. Ver. 1.1, Bundesamt für Sicherheit in der Informationstechnik, Bundeskriminalamt, Fraunhofer Institut für Graphische Datenverarbeitung,
http://www.bsi.de/literat/studien/BioFinger/BioFinger_I_I.pdf,
gesehen am 26.01.2007.
- [44] Behrens, M.: *Biometrische Identifikation. Grundlagen, Verfahren, Perspektiven*. 1. Aufl., Vieweg Verlag, Wiesbaden, 2001.
- [45] Todt, M.; Bauer, P.: *Benutzer-Authentifizierung*. Bacher Systems, Wien, Oktober 2004,
http://www.bacher.at/download/loesungen/bacher.at_benutzer-authentifizierung.pdf,
gesehen am 26.01.2007.

- [46] Lafon, Y: *Web Services Activity*.
W3C. World Wide Web Consortium (Hg). 2007,
<http://www.w3.org/2002/ws/>,
gesehen am 22.05.2007.
- [47] Booth, D.; Haas, H.; McCabe, F.; Newcomer, E.; Champion, M.; Ferris, C.;
Orchard, D.: *Web Services Architecture. W3C Working Group Note 11
February 2004*. W3C. World Wide Web Consortium (Hg). 2004,
<http://www.w3.org/TR/ws-arch/>,
gesehen am 22.05.2007.
- [48] Burke, B.; Monson-Haefel, R.: *Enterprise JavaBeans 3.0. 5th Edition*,
O'Reilly & Associates, 2006.
- [49] Wyke-Smith, C.: *Codin' for the Web: A Designer's Guide to Developing
Dynamic Web Sites*, New Riders, 2006.
- [50] Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M.: *Pattern-
Oriented Software Architecture, Vol. 1, A System of Patterns*. Wiley, 1996.
- [51] Schmidt, D.; Stal, M.; Rohnert, H.; Buschmann, F.: *Pattern-Oriented Software
Architecture, Vol. 2, Patterns for Concurrent and Networked Objects*.
Wiley, 2000.
- [52] Dewire, D.: *Thin Clients: Web-based Client/Server Architecture and
Applications*. McGraw-Hill Education, 1998.
- [53] Bien, A.: *Enterprise Architekturen. Leitfaden für effiziente Software-
Entwicklung*. 1. Aufl., Software & Support Verlag, Frankfurt, 2006.
- [54] Cardwell, L.: *AJAX – Bridging the Thin-Client Performance Gap*. IronSpeed,
[http://www.ironspeed.com/articles/AJAX-Bridging%20the%20Thin-
Client%20Performance%20Gap/Article.aspx](http://www.ironspeed.com/articles/AJAX-Bridging%20the%20Thin-Client%20Performance%20Gap/Article.aspx),
gesehen am 26.01.2007.

- [55] Fielding, R.; Gettys, J.; Mogul, J.; Frystyk, H.; Masinter, L.; Leach, P.; Berners-Lee, T.: *RFC 2616 - Hypertext Transfer Protocol - - HTTP/1.1*. Network Working Group (Hg), 1999, <http://tools.ietf.org/html/rfc2616>, gesehen am 26.01.2007.
- [56] Rescorla, E.; *RFC 2818 - HTTP Over TLS*. Network Working Group (Hg), 2000, <http://tools.ietf.org/html/rfc2818>, gesehen am 26.01.2007.
- [57] Dierks, T.; Allen, C.: *The TLS Protocol Version 1.0*. Network Working Group (Hg), 1999, <http://tools.ietf.org/html/rfc2246>, gesehen am 26.01.2007.
- [58] Raggett, D.; Le Hors, A.; Jacobs, I.: *HTML 4.01 Specification*. W3C. World Wide Web Consortium (Hg), 1999, <http://www.w3.org/TR/html4/>, gesehen am 26.01.2007.
- [59] Bray, T.; Paoli, J.; Sperberg-McQueen, C.; Maler, E.; Yergeau, F.; Cowan, J.: *Extensible Markup Language (XML) 1.1 (Second Edition)*. W3C Recommendation 16 August 2006, edited in place 29 September 2006. W3C. World Wide Web Consortium (Hg), 2006, <http://www.w3.org/TR/2006/REC-xml11-20060816/>, gesehen am 26.01.2007.
- [60] Bos, B.; Celik, T.; Hickson, I.; Lie, H.: *Cascading Style Sheets, level 2 revision 1. CSS 2.1 Specification*. W3C Working Draft 06 November 2006. W3C. World Wide Web Consortium (Hg), 2006, <http://www.w3.org/TR/CSS21/>, gesehen am 26.01.2007.
- [61] Zakas, N.: *Professional JavaScript for Web Developers*. John Wiley & Sons, 2005.

- [62] Ambler, S.: *Mapping Objects to Relational Databases: O/R Mapping In Detail*. Ambysoft,
<http://www.agiledata.org/essays/mappingObjects.html>
gesehen am 26.01.2007.
- [63] Sun Microsystems (Hg): *Java Platform, Standard Edition 6. API Specification*.
2006,
<http://java.sun.com/javase/6/docs/api/>,
gesehen am 26.01.2007
- [64] Arnold, K.; Gosling, J., Holmes, D.: *The Java Programming Language*.
4. Aufl., Addison Wesley Longman, Amsterdam, 2005
- [65] Fowler, M.: *Refactoring. Oder wie Sie das Design vorhandener Software verbessern*. 1. Aufl., Addison Wesley, München, 2005.
- [66] Meyers, S.: *Effektiv C++ programmieren*. 1. Aufl., Addison-Wesley, München,
2005
- [67] Carlson, D.: *Eclipse Distilled*. Addison Wesley Longman, Amsterdam, 2005.
- [68] Daum, B.: *Rich-Client-Entwicklung mit Eclipse 3.2. Anwendungen entwickeln mit der Rich Client Platform*. 2. Auflage, Dpunkt Verlag, Heidelberg, 2006.
- [69] The Apache Software Foundation (Hg): *Cocoon Features*.
In: The Apache Cocoon Project,
<http://cocoon.apache.org/2.1/features.html>,
gesehen am 26.01.2007.
- [70] Langham, M.; Ziegeler, C.: *Cocoon: Building XML Applications*. 1. Aufl.,
New Riders, Indianapolis, 2003.
- [71] Moczar, L.; Aston, J.: *Cocoon Developer's Handbook*. Sams Publishing,
Indianapolis, 2003.
- [72] Schatten, A., Pötz, R.: *Apache Cocoon. Webapplikationen, Systemintegration und Cross Publishing mit dem XML-Framework*.
1. Aufl., Dpunkt Verlag, Heidelberg, 2006.

- [73] Leung, T.: *Professional XML Development with Apache Tools: Xerces, Xalan, Fop, Cocoon, Axis, Xindice*. John Wiley & Sons, 2005.
- [74] Hürsch, W.; Lopes, C.: *Separation of concerns*. Technical Report NU-CCS-95-03, Northeastern University, Boston, 1995.
- [75] IBM Research (Hg): *Multi-Dimensional Separation of Concerns: An Overview*.
<http://www.research.ibm.com/hyperspace/MDSOC.htm>,
gesehen am 26.01.2007.
- [76] The Apache Software Foundation (Hg): *Introducing Apache Cocoon*. In: The Apache Cocoon Project,
<http://cocoon.apache.org/2.1/introduction.html>,
gesehen am 26.01.2007.
- [77] Andrews, M.: *Sun Developer Network (SDN): Story of a Servlet: An Instant Tutorial*. Sun Microsystems,
<http://java.sun.com/products/servlet/articles/tutorial/>,
gesehen am 26.01.2007.
- [78] Sun Microsystems (Hg): *Sun Developer Network (SDN): The Java Servlet API White Paper*.
<http://java.sun.com/products/servlet/whitepaper.html>
gesehen am 26.01.2007.
- [79] Perry, B.: *Java Servlet and JSP Cookbook*. 1. Aufl., O'Reilly Media, 2004.
- [80] Kurniawan, B.; Deck, P.: *How Tomcat Works: A Guide to Developing Your Own Java Servlet Container*. Brainysoftware.com, Vancouver, 2004.
- [81] Mort Bay Consulting (Hg): *jetty6 – Jetty WebServer*. In: Jetty,
<http://jetty.mortbay.org/>,
gesehen am 26.01.2007.

- [82] The Apache Software Foundation (Hg): *Apache Tomcat 6.0*.
In: Apache Tomcat 6.0,
<http://tomcat.apache.org/tomcat-6.0-doc/introduction.html>,
gesehen am 26.01.2007.
- [83] The Apache Software Foundation (Hg): *Apache Module mod_proxy.*,
In: Apache HTTP Server,
http://httpd.apache.org/docs/2.0/mod/mod_proxy.html,
gesehen am 26.01.2007.
- [84] Kersken, S.: *Apache 2*. 2. Aufl., Galileo Press, Bonn, 2005.
- [85] Belapurkar, A.: *Use Continuations to develop complex Web applications*.
IBM developerWorks, 2004,
<http://www-128.ibm.com/developerworks/library/j-contin.html>,
gesehen am 26.01.2007.
- [86] The Apache Software Foundation (Hg): *Cocoon Forms Introduction*.
In: The Apache Cocoon Project,
<http://cocoon.apache.org/2.1/userdocs/basics/index.html>,
gesehen am 26.01.2007.
- [87] Brownell, D.: *SAX2*. 1. Aufl., O'Reilly Media, 2002.
- [88] Berners-Lee, T.; Fielding, R.; Masinter, L.: *RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax*. Network Working Group (Hg), 2000,
<http://tools.ietf.org/html/rfc3986>,
gesehen am 26.01.2007.
- [89] Fowler, M.: *Inversion of Control Containers and the Dependency Injection pattern*. 2004,
<http://www.martinfowler.com/articles/injection.html>,
gesehen am 26.01.2007.
- [90] The Apache Software Foundation (Hg): *What is excalibur?*.
In: Apache Excalibur,
<http://excalibur.apache.org/index.html>,
gesehen am 26.01.2007.

- [91] Eagle, M.: *Wiring Your Web Application with Open Source Java*. O'Reilly OnJava.com, 2004,
<http://www.onjava.com/pub/a/onjava/2004/04/07/wiringwebapps.html>,
gesehen am 26.01.2007.
- [92] Google (Hg): *Google Web Toolkit - Build AJAX apps in the Java language*. In: Google Code, 2007,
<http://code.google.com/webtoolkit/>,
gesehen am 26.01.2007.
- [93] Hanson, R.; Tacy, A.: *GWT in Action: Easy Ajax with Google Web Toolkit*. Manning Publications Co., Greenwich, 2007.
- [94] Chaganti, P.: *Google Web Toolkit: GWT Java AJAX Programming*. Packt Publishing, Birmingham, 2007.
- [95] *Apache License, Version 2.0*. The Apache Software Foundation,
<http://www.apache.org/licenses/LICENSE-2.0.html>,
gesehen am 26.01.2007.
- [96] Google (Hg): *Developer Guide*. In: Google Code, 2007,
<http://code.google.com/webtoolkit/documentation/com.google.gwt.doc.DeveloperGuide.html>,
gesehen am 26.01.2007.
- [97] *HIBERNATE - Relational Persistence for Idiomatic Java. Hibernate Reference Documentation. 3.2.1*. Red Hat Middleware, 2006,
http://www.hibernate.org/hib_docs/v3/reference/en/pdf/hibernate_reference.pdf,
gesehen am 26.01.2007.
- [98] Bauer, C.; King, G.: *Hibernate in Action. Practical Object/Relational Mapping*. Manning, 2004.
- [99] Bauer, C.; King, G.: *Java Persistence with Hibernate*. Manning, 2006.

- [100] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: *Entwurfsmuster. Elemente wieder verwendbarer objektorientierter Software*. Addison Wesley, München, 2004.
- [101] Link, J.: *Softwaretests mit JUnit*. 2. überarb. u. erw. Aufl., Dpunkt Verlag, Heidelberg, 2005.
- [102] Hunt, A.; Thomas, D.: *Unit-Tests mit JUnit*. Hanser, München, 2004.
- [103] *ReActor for Process heat, Hydrogen And Electricity generation. Integrated Project of the 6th Framework Programme of the Euratom Community*.
<http://www.rafael-project.org/index.html>,
gesehen am 21.05.2007.
- [104] Lohnert, G.H.; Reutler, H.: *The Modular High-Temperature Reactor*. In Nuclear Technology, Vol. 62, p. 22-30, American Nuclear Society, Illinois, 1983.
- [105] *ENEN. European Nuclear Education Network*.
<https://www.enen-assoc.org/>,
gesehen am 22.05.2007.
- [106] Weigele, M.; Achenbach, J.; Piater, A.; Schmidt, F.; Schulz, A.; Sucic, D.; Obrecht, R.; Weimer, S.; Bechtler, R.: *KFÜ-ABR – Entwicklung eines ABR-Research Systems*. In: UIS Baden-Württemberg. Projekt AJA. Anwendung JAVA-basierter und anderer leistungsfähiger Lösungen. Phase V-2003. FZKA-6950, Mayer-Föll, R.; Keitel, A.; Geiger, W. (Hg), Wissenschaftliche Berichte des Forschungszentrums Karlsruhe, IAI, 2004.
- [107] Krass, C.; Achenbach, J.; Scheuermann, W.; Obrecht, R.; Pohl, H.: *KFÜ-ABR - Untersuchung möglicher Erweiterungen des Anwendungsbereichs von ABR-Research hinsichtlich Diagnose-/Prognoseausbreitungsrechnungen und Ausbreitung in kleinräumigen Gebieten*. In: UIS Baden-Württemberg. F+E-Vorhaben KEWA. Phase I – 2005/06. FZKA-7250, Mayer-Föll, R.; Keitel, A.; Geiger, W. (Hg), Wissenschaftliche Berichte des Forschungszentrums Karlsruhe, IAI, 2006.

- [108] Piater, A.; Scheuermann, W.; Krass, C.; Wagner, D.; Obrecht, R.; Pohl, H.: *Anbindung an die zentrale Datenhaltung der Kernreaktor-Fernüberwachung Baden-Württemberg zur Durchführung von Prognoserechnungen*. In: UIS Baden-Württemberg. F+E-Vorhaben KEWA. Phase II – 2006/07. FZKA-7350, Mayer-Föll, R; Keitel, A.; Geiger, W. (Hg), Wissenschaftliche Berichte des Forschungszentrums Karlsruhe, IAI, 2007.
- [109] Google (Hg): *Google Maps API*. In: Google MapsCode, <http://www.google.com/apis/maps/>,
gesehen am 21.05.2007.
- [110] Brown, M.: *Hacking Google Maps and Google Earth*. 1. Aufl., John Wiley & Sons, 2006.
- [111] *NEPTUNO CS*.
<http://www.neptuno-cs.de/>,
gesehen am 21.05.2007.
- [112] *GRS-FBW Datenbank der Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH*.
<https://www.grs-fbw.de/>,
gesehen am 21.05.2007.
- [113] Piater, A.; Lurk, A.: *Role-Based View Control in Web-Based Simulation Environments*. In: Proceedings of the 6th EUROSIM Congress on Modelling and Simulation. Ljubljana, 2007.

Institut für Kernenergetik und
Energiesysteme

Universität Stuttgart

Pfaffenwaldring 31

D-70550 Stuttgart

