

Andreas Heil

Anwendungsentwicklung für Intelligente Umgebungen im Web Engineering

RESEARCH



Springer Vieweg

Anwendungsentwicklung für Intelligente Umgebungen im Web Engineering

Andreas Heil

Anwendungsentwick- lung für Intelligente Umgebungen im Web Engineering



Springer Vieweg

RESEARCH

Andreas Heil
Linkenheim, Deutschland

Vollständiger Abdruck der von der Fakultät für Informatik
der Technischen Universität Chemnitz genehmigten Dissertation.

ISBN 978-3-8348-2550-6

ISBN 978-3-8348-2551-3 (eBook)

DOI 10.1007/978-3-8348-2551-3

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Vieweg

© Springer Fachmedien Wiesbaden 2012

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Vieweg ist eine Marke von Springer DE. Springer DE ist Teil der Fachverlagsgruppe Springer Science+Business Media
www.springer-vieweg.de

Vorwort

Jenseits des herkömmlichen Software Engineerings existiert die noch recht junge Disziplin des Web Engineerings, die sich der ingenieurmäßigen Entwicklung, Wartung und dem Betrieb von Web-Anwendungen widmet. Da der Einsatz von Web-Technologien für die Umsetzung der vorliegenden Arbeit wie geschaffen erschien, bot es sich an, die Arbeit in genau diesem Forschungsfeld anzusiedeln. Gefördert durch ein Stipendium im Rahmen des „Microsoft Research PhD Scholarship Programme“ entstand die vorliegende Arbeit während meiner Tätigkeit als wissenschaftlicher Mitarbeiter und Promotionsstudent an der Professur für Verteilte und Selbstorganisierende Rechnersysteme der Fakultät für Informatik an der Technischen Universität Chemnitz.

Ein persönliches Ziel bestand darin, bereits existierende Ansätze aufzugreifen und vorherige Forschungsergebnisse und Erkenntnisse auszubauen. So entwickelt diese Arbeit zum einen die Idee des WebComposition-Ansatzes weiter und soll zum anderen als Ausgangspunkt für zukünftige Arbeiten dienen. Im Vordergrund stand dabei die praktische Anwendbarkeit der erarbeiteten Ergebnisse. Insbesondere ist dies mit dem im Rahmen dieser Arbeit entwickelten Data Grid Service gelungen, der bereits zur Drucklegung dieser Arbeit in mehreren Forschungsprojekten eingesetzt und aktiv weiterentwickelt wurde. Weitere konkrete Ergebnisse dieser Arbeit wurden zusammen mit der Firma Microsoft beispielsweise in Form zweier Patentschriften für die zukünftige industrielle Verwendung aufbereitet.

Die vorliegende Arbeit unterscheidet sich bewusst in der Struktur von der typischen Dissertationsschrift. Jeder einzelne Teilbereich wird gleichermaßen aufbereitet und letztendlich in einem ganzheitlichen Prozess zusammengeführt. Zusammengefasst wendet sich das Buch nicht nur an den Forscher, sondern auch an den Lehrenden, den Lernenden als auch den Entwickler, der sich mit der Thematik des Web Engineerings auseinandersetzen möchte. Vorausgesetzt ist jedoch ein gewisses Maß an Kenntnissen der Informatik, des Software Engineerings und im Umgang mit Web-Technologien.

Wie bei allen umfangreichen Projekten wäre diese Arbeit nie ohne die Unterstützung zahlreicher Helfer zustande gekommen. Mein besonderer Dank gilt meinem Doktorvater Prof. Martin Gaedke, der nicht nur ein Mentor war, sondern über die Jahre auch zu einem guten Freund wurde. Die zahlreichen anregenden

Diskussionen waren immer eine Quelle neuer Ideen und Ansätze für spannende Arbeiten. Ebenso bedanke ich mich bei Prof. Wolfram Hardt für die Übernahme des Koreferats meiner Arbeit.

Auch dem Kollegiat der Professur, den Herren Dr. Jörg Anders, Ralf Sontag und Jens Wegener, gilt mein Dank für die freundliche Aufnahme in ihren Reihen. Besonders Herrn Dr. Johannes Meinecke danke ich für die wertvolle Zusammenarbeit sowohl an der Technischen Universität Chemnitz als auch bereits zuvor an der Universität Karlsruhe (TH). Für ihre wertvollen Beiträge als Studienarbeit, Diplomanden und wissenschaftliche Hilfskräfte möchte ich allen beteiligten Studenten meinen Dank aussprechen, welche durch ihre Beiträge die Entstehung dieser Arbeit unterstützt haben. Besonders Herrn Sebastian Brandt bin ich für sein außerordentliches Engagement dankbar.

Herr Tomasz Naumowicz von der Freien Universität Berlin danke ich für die Unterstützung bei der Umsetzung des ScatterWeb-Projektes. Den Herren Hartmut Knecht und Dirk Haizmann der Firma fischertechnik gilt mein Dank für die Erfüllung eines Kindheitstraumes, da sie die Umsetzung des e-Homes in den Produktionsstätten fischertechniks erst möglich gemacht haben.

Herrn Dr. Alexander Brändle gilt mein Dank, da er mir im Rahmen meiner Tätigkeit bei der Firma Microsoft Research uneingeschränkten Freiraum eingeräumt hat, meine Forschung durchzuführen und meine Arbeit fertigzustellen. Als Vorbild sowohl beruflich als auch menschlich hat er maßgeblich zu meinem Werdegang beigetragen.

Für die zahlreichen Stunden Korrekturlesen der Dissertationsschrift möchte ich mich bei meinen ehemaligen Kommilitonen und langjährigen Freunden Herrn Michael Köhnlein und Herrn Matthias Friedrich bedanken.

Meiner Freundin Katharina Becker gilt ein besonderes „Dankefein“, da sie mir während der Endphase meiner Arbeit motivierend zur Seite stand und durch ihre Unterstützung maßgeblich dazu beigetragen hat, dieses Projekt letztendlich zu einem erfolgreichen Abschluss zu führen.

Mein letzter Dank gilt meiner Familie, insbesondere meinen Eltern Hildburg und Frank Heil, welche mich in meinem Werdegang stets vorbehaltlos unterstützen und meinen beruflichen Weg von klein auf ermöglicht haben.

Andreas Heil

Inhaltsverzeichnis

Vorwort.....	V
Abkürzungsverzeichnis	XI
Abbildungsverzeichnis.....	XV
Tabellenverzeichnis	XIX
1 Einleitung.....	1
1.1 Ausgangssituation	1
1.2 Problemstellung	3
1.3 Motivation.....	5
1.4 Beiträge und Einordnung	7
1.5 Abgrenzung der Arbeit.....	9
1.6 Gliederung der Abhandlung.....	9
2 Grundlagen und Prinzipien	13
2.1 Ausgangssituation	13
2.2 Problemstellung	15
2.3 Web-Technologien für Intelligente Umgebungen.....	17
2.4 Prinzipien	24
2.5 Zusammenfassung.....	34
3 Vorgehensmodelle	37
3.1 Ausgangssituation	37
3.2 Problemstellung	39
3.3 Stand der Technik	41
3.4 WebComposition/WBS.....	64
3.5 Zusammenfassung.....	76
4 Methoden	79
4.1 Ausgangssituation.....	79
4.2 Problemstellung	81
4.3 Stand der Technik	82

4.4	WebComposition/Test	92
4.5	Zusammenfassung	121
5	Werkzeuge.....	123
5.1	Ausgangssituation	123
5.2	Problemstellung	125
5.3	Stand der Technik	127
5.4	WebComposition/DGS	138
5.5	Zusammenfassung	169
6	Formalismen.....	171
6.1	Ausgangssituation	171
6.2	Problemstellung	172
6.3	Stand der Technik	173
6.4	WebComposition/WCS.....	185
6.5	Zusammenfassung	198
7	Umsetzung und Bewertung.....	201
7.1	Praktische Umsetzung des Vorgehensmodells.....	201
7.2	Anwendungsszenario e-Home	204
7.3	Anwendungsszenario Sensornetzwerk.....	206
7.4	Anwendungsszenario Unternehmensdaten	209
7.5	Bewertung.....	212
7.6	Zusammenfassung	216
8	Zusammenfassung und Ausblicke.....	217
8.1	Ergebnisse der Arbeit.....	217
8.2	Weiterführende Arbeiten	220
A.	Ontologien und Schemas.....	223
A.1	Publish/Subscribe Ontologie.....	223
A.2	SA-REST Transformation	228
A.3	Kombinierte XHTML-SA-REST-Transformation.....	232
B.	Formale Darstellung.....	237
B.1	Kalkül Primitiven.....	237
B.2	WAM Modellierungselemente.....	238
B.3	WAM Ontologie	241

B.4 Bausteine für die formale Berechnung.....	247
C. Popfly Mashup	255
C.1 Popfly Skripte	255
C.2 Popfly Mash Block.....	257
Literaturverzeichnis	259
Index	299

Abkürzungsverzeichnis

ACP	Algebra of Communicating Processes
ADFS	Active Directory Federation Services
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ATC	Air Traffic Control
ATOM	Atom Publishing Protocol
AWE	Agile Web Engineering Process
BPEL	Business Process Execution Language
CBWE	Component-based Web Engineering
CDIS	CCF Display Information System
CERN	Heute European Organization for Nuclear Research, aus dem Französischen Organisation Européenne pour la Recherche Nucléaire, ursprünglich Conseil Européen pour la Recherche Nucléaire benannt
CoBrA	Context Broker Architecture
CORBA	Common Object Request Broker
COTS	commercial off-the-shelf
CPU	Central Processing Unit
CRUD	Create, Read, Updaten und Delete als grundlegende Menge von Operationen zur Datenmanipulation
CSP	Communicating Sequential Processes
CSS	Calculus of Communicating Systems
ECMA	Heute Ecma International European Association for Standardizing Information and Communication Systems, ursprünglich unter der Bezeichnung European Computer Manufacturers Association geführt
EIB	Europäischer Installationsbus
ER	Entity-Relationship Model
eW3DT	Extended World Wide Web Design Technique
FLiP	Fusebox Lifecycle Process

FOAF	Friend of a Friend
EXIF	Exchangeable Image File
GIS	Geographische Informationssysteme
GRDDL	Gleaning Resource Descriptions from Dialects of Languages
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HTTP-URI	Ein im Zusammenhang mit dem Transportprotokoll HTTP verwendeter Uniform Resource Identifier
ICWE	International Conference on Web Engineering
idFS	Identity Federation System
IIOF	Internet Inter-Operability Protocol
IP	Identity Provider
iROS	Interactive Room Operating System
ISM	Industrial, Scientific and Medical Band
IT	Information Technology
ITIL	IT Infrastructure Library
ITX	Internet Transaction Identifier
JOD	JavaScript on Demand
JSON	JavaScript Object Notation
LON	Local Operating Network
MDA	Model-driven Architecture
MOF	Microsoft Operations Framework
MSF	Microsoft Solutions Framework
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
OMT	Object-Modeling Technique
OOHDM	Object-Oriented Hypermedia Design Method
OpenUP	Open Unified Process

OSGi	Ursprünglich Open Services Gateway Initiative, jetzt unter der Bezeichnung OSGi Alliance geführte Standardisierungsorganisation
OWL	Web Ontology Language
PC	Personal Computer
PMBOK	Project Management Book of Knowledge
POX	Plain Old XML
RDF	Resource Description Framework
RDFa	RDF in HTML attributes
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
RFID	Radio Frequency Identification
RSS	Eine Familie von auf XML-basierten Formaten zur Aggregation von Informationen mit versionsabhängigen Bedeutungen: RDF Site Summary (RSS 0.90), Rich Site Summary (RSS 0.91), Really Simple Syndication (RSS 2.0.1)
RUP	Rational Unified Process
SAM	Sensor-Actuator Module
SAWSDL	Semantic Annotations for WSDL and XML Schema
SHDM	Semantic Hypermedia Design Method
SOA	Service Oriented Architecture
SOAP	Ursprünglich Simple Object Access Protocol, jetzt eigenständiger Begriff
SPS	Speicherprogrammierbare Steuerung
STS	Security Token Service
TCA	Transaction Costs Analysis
TCP/IP	Transmission Control Protocol/Internet Protocol
TSCP	Triple Space Transfer Protocol
UDP	User Datagram Protocol
UIX	User Interface Experience
UML	Unified Modeling Language
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier

URL	Uniform Resource Locator
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WAM	WebComposition Architecture Model
WCAG	Web Content Accessibility Guidelines
WCF	Windows Communication Framework
WCS	WebComposition Concurrency System
Web	World Wide Web
Web-GIS	Web-basierte geographische Informationssysteme
WfMS	Workflow-Managementsystem
WSA	Web Service Architecture
WSDL	Web Service Description Language
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XMI	XML Metadata Interchange
XML	Extensible Markup Language
XML-RPC	Ein über HTTP übermittelter, in XML kodierter Remote Procedure Call
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations
XPath	XML Path Language

Abbildungsverzeichnis

Abbildung 1:	Konstruktive Maßnahmen der ganzheitlichen Betrachtung von Entwicklungsprozessen für Softwaresysteme	5
Abbildung 2:	Daten- und Steuerungsfluss innerhalb einer Gebäudeautomation	14
Abbildung 3:	Einsatz von Web-Technologien in Intelligenten Umgebungen	17
Abbildung 4:	Darstellung von RDF Daten als Graph.	22
Abbildung 5:	Evolutionsdimensionen von Web-Anwendungen für Intelligente Umgebungen	26
Abbildung 6:	Horizontale und vertikale Wiederverwendung von Softwareartefakten	27
Abbildung 7:	Prinzipien und ihr Zusammenhang im Rahmen konstruktiver Elemente der ingenieurmäßigen Entwicklung	34
Abbildung 8:	Unterschiedliche Entwicklungsstufen von Teilsystemen und gegenseitige Beeinflussung durch deren Weiterentwicklung ...	39
Abbildung 9:	Historische Entwicklung von Modellierungssprachen im Web Engineering	42
Abbildung 10:	Am Spiralmodell orientiertes Vorgehensmodell im WebML Entwicklungsprozess.....	48
Abbildung 11:	Lebenszyklus nach der Relationship Management Methodology	50
Abbildung 12:	Evolutionsspirale auf Basis des WebComposition Vorgehensmodells.....	52
Abbildung 13:	Produktlebenszyklus und Meilensteine nach Microsoft Solution Framework.....	55
Abbildung 14:	Integriertes Microsoft Solutions Framework IT-Lebenszyklus Vorgehensmodell	56
Abbildung 15:	Produktlebenszyklus und Meilensteine nach dem OpenUP Modell.....	58
Abbildung 16:	Produktzustandsmodell nach dem V-Modell XT.....	59
Abbildung 17:	Produktlebenszyklus nach dem V-Modell XT	60
Abbildung 18:	WebComposition/WBS Ansatz zur Anwendungsentwicklung in Web-basierten Systemen	65

Abbildung 19: Tailoring-Aspekte zur Anwendungsentwicklung für Intelligente Umgebungen im Web Engineering.....	66
Abbildung 20: Tailoring zur Strukturierung der Anwendungsentwicklung im WebComposition/WBS	67
Abbildung 21: Konfigurationszustandsmodell des WebComposition/WBS Ansatzes.....	69
Abbildung 22: WebComposition Vorgehensmodellmodell zur Entwicklung Web-basierter Systeme	70
Abbildung 23: Nutzung von Prozessartefakten in der Anwendungsentwicklung und im Betrieb mittels eines Wiederverwendungsrepositoriums	75
Abbildung 24: Nichtineinandergreifendes Testen unterschiedlicher Systemkomponenten innerhalb Web-basierter Systeme.....	80
Abbildung 25: Testen als konstruktive Maßnahme bestehen aus Verifikation und Validierung.....	85
Abbildung 26: Test-Dimensionen im WebComposition/Test Model	93
Abbildung 27: Testzyklus zum funktionalen Testen innerhalb des WebComposition/WBS Ansatzes	97
Abbildung 28: Vereinfachte, auf (Fielding, 2000) basierende Darstellung des REST-Architekturstils	100
Abbildung 29: Zu testende Schritte bei Ressourcen-basierten Abfragen.....	107
Abbildung 30: Fehlertypen nach (Binder, 1999) in der herkömmlichen Softwareentwicklung	108
Abbildung 31: Beobachten eines Fehlers aufgrund eines Ubiquitären Versagens.....	109
Abbildung 32: WebComposition Risikomodel.....	114
Abbildung 33: Risikobewertung anhand des WebComposition Riskomodells	117
Abbildung 34: Identifikation von Testprioritäten	118
Abbildung 35: Prozess zum permanenten Risikomanagement.....	119
Abbildung 36: Auf (Wache, 2008) basierendes Web-GIS zum Einsatz im Katastrophenschutz.....	124
Abbildung 37: Vereinheitlichung im Umgang mit Software und Daten im Rahmen der Anwendungsentwicklung für Intelligente Umgebungen.....	127
Abbildung 38: Middleware zur Anbindung physischer Ressourcen.....	128
Abbildung 39: Zugriff auf Ressourcen mittels Protokoll-basierter Ansätze ...	131

Abbildung 40: Repräsentation von Ressourcen mittels anwendungsspezifischer Container	132
Abbildung 41: Zugriff auf Ressourcen über Datenbank-getriebene Ansätze ...	134
Abbildung 42: Unterschiedliche Anforderung an das Lösungskonzept	140
Abbildung 43: Ressourcen Übermittlung im Web	141
Abbildung 44: Web Charakterisierung nach (Lavoie und Nielsen, 1999)	142
Abbildung 45: Vereinfachtes URI-Venn-Diagramm	144
Abbildung 46: Gegenüberstellung von HTTP-URI und URN	145
Abbildung 47: Komponentendiagramm der WebComposition/DGS Sub- Komponenten	147
Abbildung 48: Verbunddienst basierend auf einer WebComposition/DGS Komponente	148
Abbildung 49: Das WebComposition/DGS-Ressourcenmodell und dessen semantische Verknüpfungen	149
Abbildung 50: Verwendung von Semantischer URI zur Beschreibung von Ressourcen	150
Abbildung 51: 303 URI Weiterleitung	152
Abbildung 52: WebComposition/DGS Information Stack	153
Abbildung 53: Daten Provenance mittels implizit generierter Metainformation im WebComposition/DGS-Rahmenwerk	154
Abbildung 54: Topic-basierter Publish/Subscribe-Mechanismus im WebComposition/DGS-Rahmenwerk	157
Abbildung 55: Auftreten von Ereignissen innerhalb des Subscription Bus	159
Abbildung 56: Einsatz einer Internet Transaction Identifier (ITX) zur Vermeidung von Lost Updates bei der Manipulation von Ressourcen	161
Abbildung 57: Verbunddienst auf Basis der WebComposition/DGS- Komponente zur Realisierung von Navigationsstrukturen	164
Abbildung 58: Auf Basis eines Navigationsdienstes generiertes Navigationsmuster	165
Abbildung 59: Anbindung der SA-REST-Erweiterung an die Extension Pipeline des WebComposition/DGS-Rahmenwerks	167
Abbildung 60: Entwicklung (formaler) Beschreibungen weg vom Ist-Zustand des Systems	172
Abbildung 61: Auf (Parrow, 2001) basierendes Beispiel zum Zugriff auf physische Ressourcen	180
Abbildung 62: Grundlegende Kommunikation zwischen Prozessen im π -Kalkül	181

Abbildung 63: Ambient Kalkül am Beispiel der Active Folder basierend auf (Cardelli, 2006).....	186
Abbildung 64: Modell vernetzter Haustechnik auf Basis eines Web-basierten Systems.....	191
Abbildung 65: Modellierung Web-basierter Systeme mit Hilfe von Microsoft Visio.....	193
Abbildung 66: Hierarchische Ordnung eines WAM Modells als Ausgangspunkt zur Formalisierung des Modells.....	194
Abbildung 67: Schematische Darstellung eines WAM-Modells.....	195
Abbildung 68: Modellverifikation auf Basis des Domäneneffekts im Rahmen der Systemevolution.....	196
Abbildung 69: Auswirkung des Domäneneffekts auf Basis des formalisierten Modells.....	197
Abbildung 70: Zeitliche Entwicklung und Rückführung zusätzlicher Systemkomponenten in das WebComposition/DGS-Rahmenwerk.....	202
Abbildung 71: E-Home Modell auf Basis von fischertechnik-Bauelementen.....	205
Abbildung 72: Web-Anwendung zur Steuerung eines E-Homes in einer föderativen Umgebung.....	206
Abbildung 73: Werkzeuggestützter Entwicklungszyklus der ScatterWeb-MSB-430-Plattform.....	208
Abbildung 74: Mashup der zur Verfügung gestellten Daten auf Basis von Microsoft Popfly.....	209
Abbildung 75: Beispiel für sukzessiv wachsende Unternehmensdaten im Web.....	210
Abbildung 76: Durch eine WebComposition/DGS-Komponente verwaltete, organisatorisch gewachsene Informationsstruktur von Unternehmensdaten.....	211
Abbildung 77: Erweiterung von Metainformationen von Publikationen durch den Autor.....	212

Tabellenverzeichnis

Tabelle 1:	Methoden der Modellierung von Web-Anwendungen.....	44
Tabelle 2:	Vergleich der vorgestellten Vorgehensmodelle	61
Tabelle 3:	Merkmalsbeschreibungen und -abkürzungen.....	63
Tabelle 4:	Bewertungsschema für Vorgehensmodelle.....	64
Tabelle 5:	Vergleich Testbewusstsein im Vergleich Software Engineering (SE) zu Web Engineering (WE)	83
Tabelle 6:	Exemplarische Auswahl von Werkzeugen zum Testen Web-basierter Systeme	91
Tabelle 7:	Test-Ergebnisse des Ressourcen-orientierten Testens	106
Tabelle 8:	Abkürzungen und Bewertungsschema für Testansätze.....	137
Tabelle 9:	Übersicht über die vorgestellten datenzentrierten Ansätze zur Verwaltung von Ressourcen bezogen auf Prinzipien und Kriterien zur Integration physischer Ressourcen	138
Tabelle 10:	Abgrenzungskriterien für Ambients.....	182
Tabelle 11:	Kommunikationsmechanismen innerhalb des Ambient-Kalküls.....	183
Tabelle 12:	WAM Modellierungselemente.....	238

1 Einleitung

Die rasante technologische und informationstechnische Entwicklung im 21. Jahrhundert, vor allem bedingt durch das Aufkommen des World Wide Web, stellt gänzlich neue Herausforderungen an die Entwicklung und Integration moderner, interdisziplinär entwickelter Software. Anwendungen werden nicht mehr für einzelne, isolierte Systeme entwickelt, sondern sind oftmals Bestandteil größerer, verteilter Systeme, deren einzelne Systemkomponenten miteinander kommunizieren und Daten untereinander austauschen. Eine Ausprägung solcher verteilter Systeme stellen *Intelligente Umgebungen* dar, die aufgrund des Einsatzes von Sensoren und Aktuatoren adaptiv auf ihre Umgebung reagieren können. Eine Besonderheit dieser Art verteilter Umgebungen ist die rasche Veränderung der zugrundeliegenden Systemstruktur, beispielsweise durch das Hinzufügen oder Entfernen von Sensoren. Während die Anwender diese fortwährende Veränderung möglichst nicht bemerken sollen, stellt sie die Anwendungsentwickler vor gänzlich neue Herausforderungen und erfordert eine fortwährende Weiterentwicklung und Anpassung der Anwendungen. Im Gegensatz zur herkömmlichen Anwendungsentwicklung entstehen Anforderungen permanent durch das sich ständig ändernde Gesamtsystem und ein sich veränderndes Benutzerverhalten. Fortwährende Modifikation der Infrastruktur, hinzugefügte oder wegfallende System- und Anwendungskomponenten sowie neue Arten von Daten bedürfen einer stetigen *Evolution* der Anwendung. Die vorliegende Abhandlung beschreibt die systematische Zusammenführung von *Vorgehensmodell*, *Entwicklungsprozess* und Technologien auf Basis von *Methoden* und *Werkzeugen*, die dieser speziellen Art von Anwendungsentwicklung gerecht werden.

1.1 Ausgangssituation

Das Gebiet der *Intelligenten Umgebungen* umfasst ein multidisziplinäres Forschungsumfeld und beinhaltet eine Kombination aus Infrastrukturen, verteilten Anwendungen, Diensten, einer Vielzahl vernetzter Recheneinheiten, mobiler Endgeräte, Sensoren und Aktuatoren, angereichert mit kontextuellen Informationen. Neben dem Bestreben, die Interaktion mit Computern im alltäglichen Leben zu vereinfachen und nutzerfreundlicher zu gestalten, besteht das Ziel, die Technologie, obwohl ubiquitär, also allgegenwärtig, für den Benutzer möglichst un-

sichtbar zu halten. In diesen physischen Umgebungen verschmelzen Informations- und Kommunikationstechnologien. Geräte und Sensoren verschwinden aus dem Sichtfeld des Benutzers (Steventon und Wright, 2006). Bestimmend sind hierbei Faktoren, die die nahtlose Einbettung von Geräten in die Umgebungen zur technologischen Unterstützung und Erleichterung alltäglicher Aufgaben sowohl im Beruf als auch in der Freizeit ermöglichen (Harper, et al., 2006; Steventon und Wright, 2006). Zum jetzigen Zeitpunkt wurde die Größe der Geräte auf ein Maß reduziert, das einen nahezu transparenten Einsatz im alltäglichen Leben ermöglicht. Bereits 1991 hatte Mark Weiser im Rahmen seiner Forschungsarbeit am Palo Alto Research Center die Vision einer Technologieform entwickelt, die sprichwörtlich vor dem Auge des Benutzers verschwindet (Weiser, 1991). Die Mehrheit der damals ausformulierten Ideen bestand für einen langen Zeitraum lediglich als solche weiter. In jüngster Zeit werden jedoch mehr und mehr dieser Konzepte realisiert und halten nach und nach Einzug in das tägliche Leben.

Die Forschung im Bereich der Intelligenten Umgebungen erweist sich als äußerst vielfältig und kombiniert Forschungsbereiche aus der Softwaretechnik, dem „Ubiquitous Computing“ und „Pervasive Computing“, eingebetteter Geräte, Sensoren, der Robotik, dem Gebiet der „Ambient Intelligence“ und intelligenter Benutzerschnittstellen. Zusätzlich wird auf zahlreiche Durchbrüche innerhalb von Basistechnologien anderer Forschungsbereiche zurückgegriffen, um die in der Forschungsgemeinschaft gesteckten Ziele zu erreichen. Hierzu zählen beispielsweise die fortschreitende Miniaturisierung und die Reduzierung des Stromverbrauchs im Bereich der Mikroelektronik, verbesserte Kommunikations- und Netzwerktechnologien, aber auch neuartige Konzepte wie kontextbezogene Anwendungen und semantische Konzepte, wie sie im Umfeld der Web-Technologien entwickelt werden (IET, 2005-IET, 2008).

Die Anwendungsentwicklung im Rahmen Intelligenter Umgebungen wird durch die Einflüsse unterschiedlicher, oftmals disjunkter Disziplinen beeinflusst. Zum einen setzen Intelligente Umgebungen eine hochtechnisierte Infrastruktur voraus, um mit ihrer jeweiligen Umgebung zu interagieren, grenzen sich zum anderen jedoch vom Ubiquitous Computing oder Pervasive Computing hinsichtlich einer nicht notwendigerweise allumfassenden Technisierung der betreffenden Umgebung ab (Coen, 1998). Weitere technologische Einflüsse hinsichtlich der Anwendungsentwicklung ergeben sich in diesem Umfeld aus den Forschungsbereichen der Sensorik und Robotik, während Forschungen aus dem Bereich der Mensch-Maschine-Interaktion die Schnittstellen zwischen Mensch und computerisierter Umgebung in den Vordergrund stellen.

Da der Begriffe der Intelligenten Umgebung in der Literatur nicht eindeutig definiert ist, wird dieser angelehnt an (Aarts und Encarnac, 2006) in dieser Arbeit wie folgt verwendet:

Definition 1 – Intelligente Umgebung: Ein abgegrenzter technologischer Bereich, der mittels Softwareanwendungen, Aktuatoren und Sensoren Benutzerinteraktionen erfasst, interpretiert und den Benutzer in seinen Aktionen unterstützt und sich ändernden Benutzeranforderungen anpassen kann.

1.2 Problemstellung

Die grundlegende Problemstellung der vorliegenden Abhandlung befasst sich mit der ganzheitlichen Anwendungsentwicklung und somit dem Prozess der Softwareentwicklung für Intelligente Umgebungen. Der Verlauf der Anwendungsentwicklung für Intelligente Umgebungen wird oftmals maßgeblich durch den dominierenden Forschungsbereich oder den Zweck des jeweiligen Systems vorgegeben. Während im Bereich der Mensch-Maschine-Interaktion der Fokus auf der Optimierung der Benutzerinteraktion liegt, wird in der Hardwareentwicklung überwiegend die Anbindung der entsprechenden Geräte fokussiert. In der Softwaretechnik hingegen steht oftmals die Entwicklung neuer Kommunikationsplattformen oder sogenannter Middleware-Ansätze im Blickpunkt. Abhängig vom jeweiligen Schwerpunkt ist der Prozess der Anwendungsentwicklung für Intelligente Umgebungen somit überwiegend durch die dominierende Disziplin geprägt. Jede der Teildisziplinen deckt spezifische Aspekte der zu entwickelnden Anwendungen ab, liefert jedoch keinen direkten Beitrag zur ganzheitlichen, ingenieurmäßigen Entwicklung, die aus der zielorientierter Bereitstellung und systematischen Verwendung von Prinzipien, Methoden und Werkzeugen besteht (Balzert, 1996).

Soll eine ingenieurmäßige Entwicklung von Softwaresystemen durchgeführt werden, bedarf es etablierter Konzepte, welche einen qualitativ hochwertigen Entwicklungsverlauf sicherstellen. Während analytische Maßnahmen das Vertrauen im Rahmen der Qualitätssicherung (DIN EN ISO 8402) in ein bestehendes System erhöhen, erlaubt der Einsatz *konstruktiver Maßnahmen* die Sicherstellung sowohl spezifischer Produkt- als auch Prozesseigenschaften bereits a priori im Rahmen der Qualitätslenkung. Nach (Wallmüller, 2001) lassen sich fünf konstruktive Maßnahmen innerhalb der Anwendungsentwicklung identifi-

zieren, die eine ganzheitliche Betrachtung des Entwicklungsprozesses ermöglichen (vgl. Abbildung 1).

Um eine ganzheitliche Betrachtungsweise der Anwendungsentwicklung für Intelligente Umgebungen, unabhängig vom dominierenden Fachbereich, zu gewährleisten, lassen sich bezüglich der festgestellten konstruktiven Maßnahmen folgende Probleme identifizieren, die maßgebliche Bestandteile der Anwendungsentwicklung sind.

Grundsätze, die durch *Prinzipien* festgelegt werden, sind entscheidend für die Durchführung und somit für die Qualität des Entwicklungsprozesses. Die Identifikation von Prinzipien für die Anwendungsentwicklung Intelligenter Anwendungen stellt somit den ersten Schritt dar, einen geeigneten Entwicklungsprozess zu definieren.

Das der Anwendungsentwicklung zugrundeliegende *Vorgehensmodell* beeinflusst maßgeblich die einzelnen Phasen des Entwicklungsprozesses, den operativen Betrieb und die Weiterentwicklung der Anwendung. Die Wahl eines geeigneten Vorgehensmodells trägt somit entscheidend dazu bei, die Entwicklung innerhalb des vorgegeben zeitlichen und finanziellen Rahmens durchzuführen. Hierzu steht eine Reihe von Vorgehensmodellen sowohl aus dem industriellen, als auch aus dem Forschungsumfeld zur Auswahl. Die besondere Herausforderung liegt hierbei in der Auswahl eines Vorgehensmodells, das dem Charakter sich ständig ändernder Intelligenter Umgebungen und der eingesetzten Technologien gerecht wird.

Innerhalb eines Vorgehensmodells planmäßig durchgeführte Vorgehensweisen dienen dem systematischen Erreichen festgesetzter Ziele. Diese *Methoden* beziehen sich auf spezielle Tätigkeiten und Phasen im Produktlebenszyklus und zeigen dem Anwendungsentwickler, wie diese Ziele im zur Verfügung stehenden zeitlichen und finanziellen Rahmen erreicht werden können. Hier stellt sich unter anderem die Frage, ob und welche Methoden besonders geeignet sind, das Risiko und somit Kosten und Aufwand zu minimieren, um die Anwendungsentwicklung für Intelligente Umgebungen wirtschaftlich zu gestalten.

Die Entwicklung von *Werkzeugen* für Intelligente Umgebungen stellt einen zentralen Aspekt der Anwendungsentwicklung dar. Hierbei durchziehen Anwendungen für Intelligente Umgebungen unterschiedlichste Technologieebenen. Die gleichzeitige Verwendung zahlreicher Technologien, von Web-Technologien, sowohl im Endnutzerebereich als auch zur Kommunikation, bis hin zur hardwarenahen Programmierung von Sensoren und Aktuatoren erweist sich als ein komplexes Geflecht unterschiedlicher Technologieebenen. Ein Trend dieser Komplexität entgegenzuwirken ist die Entwicklung Middleware-basierter Werkzeuge,

die die einzelnen Technologieebenen voneinander separieren und die Anwendungsentwicklung auf den jeweiligen Fachbereich einschränken. Da die so erstellten Lösungen jedoch in der Regel für dedizierte Infrastrukturen entwickelt werden, und nur ein beschränktes Maß an Wiederverwendung ermöglichen, stellt sich die Frage, ob und wie sich mittels der konsequenten Wahl von Technologien und der Abstraktion der zugrundeliegenden Ressourcen der Entwicklungsaufwand minimieren lässt.

Mit wachsender Komplexität der zu entwickelnden Anwendungen steigt der Aufwand, spezifische Fragestellungen hinsichtlich des Verhaltens einer Anwendung zuverlässig zu beantworten, signifikant an. Die automatisierte Analyse und Verifikation, die auf Basis von *Formalisen* durchgeführt wird, ist der manuellen Prüfung hierbei in vielerlei Hinsicht überlegen. Eine offene Fragestellung ist jedoch, welcher Formalismus zur Analyse und Verifikation Intelligenter Umgebungen geeignet ist, und wie sich dieser effizient in den Prozess der Anwendungsentwicklung integrieren lässt.

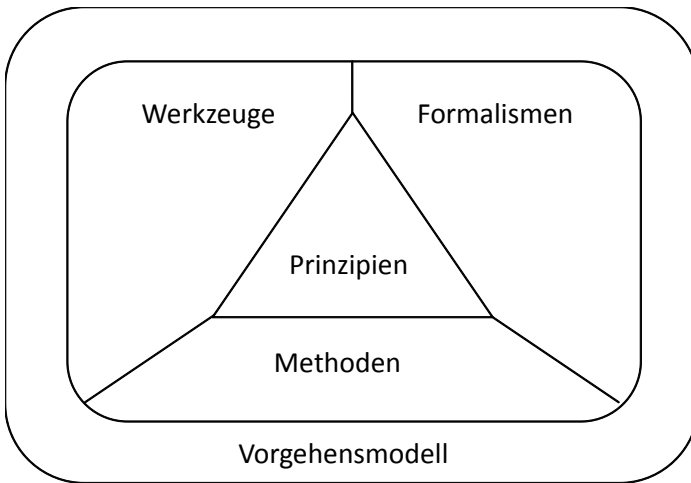


Abbildung 1: Konstruktive Maßnahmen der ganzheitlichen Betrachtung von Entwicklungsprozessen für Softwaresysteme

1.3 Motivation

Ungeachtet aller bisherigen Anstrengungen stellt sich die Frage, in welcher Form sich Anwendungsmodelle auf Basis Intelligenter Umgebungen, bestehend aus einer Vielzahl von Technologien, effektiv erstellen und weiterentwickeln lassen

(Droege, 1997). Ein erster Ansatz, der sich im größeren Rahmen über einen mehrjährigen Zeitraum mit der Thematik der Anwendungsentwicklung für Intelligente Umgebungen befasst, ist das durch die Europäische Union geförderte Hydra-Projekt (Eisenhauer, et al., 2009). Das durch eine Reihe europäischer Universitäten und Unternehmen durchgeführte Projekt zeigt unter anderem die kosteneffiziente und innovative Anwendungsentwicklung auf Basis bestehender Endgeräte. Diese Ergebnisse können zur Verbesserung der weltweiten Wettbewerbssituation des europäischen Marktes im Bereich der Software-Entwicklung beitragen.

Sowohl im Rahmen des Hydra-Projekts als auch in der Vielzahl der im Rahmen des Projekts analysierten, bereits existierenden Anwendungen und Middleware-Lösungen für Intelligente Umgebungen finden sowohl Internet- als auch Web-Technologien in einem hohen Maße Verwendung (Thestrup, et al., 2008). Der Trend des ubiquitären Web-Zugangs in Verbindung mit dem Einsatz neuester Web-Technologien bietet hier einen neuartigen Ansatz der Anwendungsentwicklung mit dem Ausgangspunkt des *World Wide Web* (WWW, Web) als Anwendungsplattform.

Das Web hat sich seit seinem Anfang von einem lokal genutzten Forschungsprojekt zu einem festen Bestandteil und einem treibenden Wirtschaftsfaktor in den vorherrschenden Industrienationen entwickelt (Berners-Lee, 2006b). Laut einer Studie von Forrester, fördern neue Technologien im Web die Entstehung neuer Kategorien von Web-Anwendungen, die Veränderung des sozialen Verhaltens, der kulturelle Entwicklung und die Entstehung neuer Geschäftsmodelle (Young, 2007b). Diese Veränderungen führen somit zu einem grundlegenden Wandel im Geschäfts- und Konsumentenverhalten. Diese weitreichende Entwicklung beeinflusst zunehmend auch die Evaluationsstrategien von Unternehmen in Bezug auf neue technologische Möglichkeiten, speziell in Bezug auf das Web. Im Jahr 2007 benutzten bereits 63-73% der IT-Manager Kennzahlen wie „Total Cost of Ownership“, „Return on Investment“, Kapitalrendite oder alternativer unternehmensinterner Kennzahlen, um Web-Technologien zu bewerten (Young, 2007a).

Auch die Notwendigkeit, auf Daten im Web zuzugreifen, diese zu interpretieren und zu manipulieren ist anhand europäischer Firmen mit einem durchschnittlichen Gesamtumsatz von mehr als 325 Millionen Euro ersichtlich. Während 75% der Unternehmen dieser Größenordnung *Service-orientierte Architekturen* (SOA) als de-facto-Standard innerhalb der Industrie erachten, werden entsprechende SOA-Strategien, in Abhängigkeit des jeweiligen Geschäftsfeld, bei bis zu 52% der Unternehmen eingesetzt (Vanson Bourne Ltd und Progress Software

Corporation, 2008). Der wesentlich jüngere Trend der sogenannten Web 2.0-Technologien konnte sich noch schneller durchsetzen. Für 50% der IT-Firmen stellt der Einsatz von Web 2.0-Technologien bereits ein elementarer Geschäftsfaktor dar, während 60% der Firmen deren Einsatz in weiteren Geschäftsbereichen in Betracht ziehen. Laut Hochrechnungen wird das auf Web 2.0-Technologien basierende Gesamtvolumen in der Wirtschaft in den kommenden Jahren stark zunehmen und bis 2013 ein globales Volumen von 4,6 Milliarden US Dollar erreichen (Young, et al., 2008).

Der gezielte Einsatz des Webs als zukünftige Anwendungsplattform zur Bereitstellung von Web-basierten Diensten, Inhalten, Daten und Metadaten wird somit zu einem zentralen Erfolgsfaktor für zukünftige Anwendungsmodelle (Abrams, 2007). Der Einsatz von Web-Technologien für die Anwendungsentwicklung Intelligenter Umgebungen erweist sich somit nicht nur aus technischer sondern auch aus wirtschaftlicher Sicht naheliegend. Während Gartner zufolge SOA und das sogenannte *föderierte Identitätsmanagement* im Web bereits großflächig Akzeptanz finden, erschließt sich auch zunehmend ein neues Feld für komplexe Ereignisverarbeitung im Web auf Basis des Einsatzes von Sensoren, wie sie in Intelligenen Umgebungen Verwendung finden (Witty und Wagner, 2003). Die fortschreitende Verfügbarkeit von ortsbezogenen Daten treibt wiederum die Entwicklung neuer, kontextsensitiver Informationstechnologien voran, die auf Web-basierten Informationsquellen, Kartenmaterial, Identitäts- und Zugriffsmagementsystemen sowie einem Geflecht aus biometrischen Geräten, Sensoren bis hin zu sogenannter intelligenter Kleidung basieren (Clark, et al., 2008).

Um den zukünftigen Entwicklungen Folge zu tragen, ist nicht nur der Einsatz von serienmäßig produzierter Hardware sondern auch die Verwendung standardisierter und jedermann zugänglicher Softwaretechnologien, wie sie durch das Web bereitgestellt werden, nötig. Hierdurch lassen sich Entwicklungszeit und Kosten, um Zugang zu den zugrundeliegenden Technologien und Diensten zu erhalten, minimieren (McCoy und Fenn, 2008).

1.4 Beiträge und Einordnung

Die vorliegende Arbeit behandelt den ganzheitlichen Ansatz zur Entwicklung von Anwendungen für Intelligente Umgebungen auf Basis des World Wide Web. Neben den softwaretechnischen Aspekten in Bezug auf Entwurf und Entwicklung des Lösungsansatzes steht der Einsatz von Web-Technologien und Standards, wie sie im World Wide Web Verwendung finden, im Vordergrund. Die

ingenieurwissenschaftliche Entwicklung des Lösungsansatzes beinhaltet neben der Softwareentwicklung verschiedene Teilaspekte der Informatik wie Modellierung, Informationsverarbeitung und Hypermedia-Ansätze. Aufgrund dieser Charakteristik lässt sich die Arbeit, die sich vorwiegend mit Entwicklung, Einsatz, Wartung und Evolution von Web-Anwendungen befasst, der Disziplin des *Web Engineering* (Deshpande, et al., 2002) zuordnen.

Ziel der Arbeit ist die Entwicklung konkreter konstruktiver Elemente für die Anwendungsentwicklung Intelligenter Umgebungen im Web Engineering. Jeder Lösungsansatz trägt jeweils einen Teil zur ganzheitlichen Betrachtung der Anwendungsentwicklung bei. Die hierzu adressierten Fragen beziehen sich direkt auf die zuvor identifizierten Problemstellungen und werden in einem jeweils separaten Kapitel behandelt:

- Welche *Prinzipien* liegen der Anwendungsentwicklung in Intelligenen Umgebungen zugrunde, denen der Entwicklungsprozess folgt?
- Wie lässt sich der evolutionäre Charakter der Anwendungsentwicklung für Intelligente Umgebungen mittels eines geeigneten *Vorgehensmodells* darstellen, um ein systematisches und strukturiertes Vorgehen für die Anwendungsentwicklung, deren Betrieb und Weiterentwicklung sicherzustellen?
- Welche *Methoden* lassen sich zur Risikominimierung in der Anwendungsentwicklung einsetzen, um den Entwicklungsprozess sowohl effizient als auch kostengünstig zu gestalten?
- Mit welchen *Werkzeugen* lässt sich eine zielorientierte Entwicklung erreichen, die den vorgegebenen Prinzipien folgt?
- Welcher *Formalismus* eignet sich, um eine differenzierte Abstraktion der Ereignisbeschreibung in Intelligenen Umgebungen dazustellen und diese methodisch zu analysieren?

Werkzeuge, Komponenten und Methoden, die in der vorliegenden Arbeit eingeführt werden, wurden im Rahmen des *WebComposition* Ansatzes entwickelt. Sie wurden über die Forschungszeit hinweg verfeinert und mittels realer Projekte evaluiert und angepasst. Die Arbeit selbst ist Teil eines fortlaufenden Forschungsvorhabens und gliedert sich wie folgt in die bestehende Arbeiten ein:

- Dissertationsschrift (Gaedke, 2000): Der hierin beschriebene WebComposition Ansatz und das entsprechende WebComposition Vorgehensmodell bil-

den die Grundlagen für die komponentenbasierte und evolutionsorientierte Entwicklung der Lösungsansätze.

- Dissertationsschrift (Nussbaumer, 2007): Dem WebComposition Ansatz folgend, liegt der Fokus dieser Arbeit auf der Komposition dienste-basierter Web-Anwendungen. Einen hohen Stellenwert nehmen hierbei die Evolution der entsprechenden Komponenten und die Ausarbeitung des ursprünglich in (Gaedke, 2000) eingeführten „Evolutionbus“ ein.
- Dissertationsschrift (Meinecke, 2008): Der WebComposition Ansatz und das Vorgehensmodell werden in dieser Arbeit hinsichtlich aktueller Sicherheitsaspekte erweitert. Dies beinhaltet vor allem die Konzepte föderierter Web-basierter Anwendungen und deren Evolution und stellt somit die Grundlage der Sicherheitsmechanismen der vorliegenden Arbeit dar.

1.5 Abgrenzung der Arbeit

Neben den in dieser Ausarbeitung behandelten Themen ergeben sich bei der Anwendungsentwicklung für Intelligente Umgebungen zahlreiche weitere Probleme, die sich dedizierten Forschungsbereichen aus der Informatik zuordnen lassen. Spezifische Fragestellungen beispielsweise aus der Sprach- und Bilderkennung, Mensch-Maschine-Interaktion, der Robotik oder der künstlichen Intelligenz, aber auch artverwandter Bereiche werden in der vorliegenden Ausarbeitung nicht weitergehend betrachtet. Zu bemerken ist jedoch, dass das in dieser Arbeit vorgestellte Vorgehensmodell die Einbeziehung entsprechender domänenspezifischer Aspekte in den Prozess der Anwendungsentwicklung explizit ermöglicht. Des weiteren werden Aspekte aus thematisch fremden Disziplinen wie der Architektur, der Elektrotechnik oder der Materialverarbeitung sowie Fragestellungen aus sozialer, rechtlicher, politischer und künstlerischer Sicht, welche mit der Problemstellung einhergehen, in der vorliegende Arbeit nicht weiter vertieft und bedürfen einer gesonderten Betrachtung.

1.6 Gliederung der Abhandlung

Die Ausarbeitung ist wie folgt gegliedert:

In Kapitel 1 wird zunächst das zentrale Thema dieser Arbeit, die Anwendungsentwicklung für Intelligente Umgebungen, dargelegt. Es werden konkrete Problemstellungen beschrieben, die in direktem Zusammenhang mit der ingenieur-

mäßigen Entwicklung von Softwaresystemen stehen. Weiterhin wird der Bezug der Anwendungsentwicklung auf Basis des World Wide Web als Anwendungsplattform motiviert. Neben der Einordnung und Abgrenzung im wissenschaftlichen Umfeld wird ein Überblick über die in dieser Arbeit beschriebenen Beiträge gegeben.

In Kapitel 2 wird zuerst eine Ausprägung Intelligenter Umgebungen anhand eines typischen Beispiels erörtert. Daraufhin werden Probleme, die sich im Zusammenhang mit einem solchen System ergeben, identifiziert. Aufbauend auf den daraus resultierenden Fragestellungen werden hiernach zunächst technologische Grundlagen eingeführt und daraufhin Prinzipien für die Anwendungsentwicklung abgeleitet, auf die im weiteren Verlauf der Arbeit immer wieder Bezug genommen wird.

Kapitel 3 befasst sich mit der Entwicklung eines geeigneten Vorgehensmodells zur Anwendungsentwicklung für Intelligente Umgebungen im Web Engineering. Hierzu werden zunächst die grundlegenden Problemstellungen vorgestellt, bevor ausgewählte Vorgehensmodelle aus den Bereichen des Web Engineering und der herkömmlichen Softwaretechnik diskutiert werden. Auf dieser Basis wird ein Vorgehensmodell entwickelt, das die spezifischen Problemstellungen adressiert.

In Kapitel 4 werden die Methoden des Testens und des Risikomanagements in Bezug auf das vorgestellte Vorgehensmodell vertieft. Die Ausgangssituation und die Problemstellung werden zunächst hinsichtlich dieses Vorgehensmodells diskutiert. Aufbauend auf dem Stand der Technik in Bezug auf Testmethoden wird ein Testmodell entwickelt, das sich in das zuvor vorgestellte Vorgehensmodell integriert und dieses um ein permanentes Risikomanagement ergänzt.

Kapitel 5 beschreibt die technologischen Konzepte des im Rahmen dieser Arbeit entstanden WebComposition/DGS-Rahmenwerkes. Zunächst werden hierzu typische Fragestellungen erläutert, die im Zusammenhang mit einer für die Anwendungsentwicklung für Intelligente Umgebungen nutzbaren Werkzeugkomponente stehen. Nach einer Analyse existierender Klassen von Werkzeugkomponenten wird das WebComposition/DGS-Rahmenwerk vorgestellt und dessen technische Realisierung diskutiert.

In Kapitel 6 wird zunächst der mögliche Einsatz formaler Methoden zur Beschreibung Intelligenter Umgebungen anhand des in dieser Arbeit entwickelten Vorgehensmodells vorgestellt. Auf Basis entsprechender Problemstellungen wird daraufhin ein Formalismus entwickelt, der insbesondere den Anforderungen Intelligenter Umgebungen gerecht wird und auf Methoden des Web Engineering basiert.

Kapitel 7 dokumentiert die Tragfähigkeit der vorgestellten Ansätze auf Basis einer Reihe domänenspezifischer Realisierungen. Hierzu werden gleichermaßen die entwickelten Vorgehensweisen und Werkzeuge eingesetzt. Dabei wird auf deren Einsatz und Wiederverwertung im Rahmen unterschiedlicher an der Professur für Verteilte und Selbstorganisierende Systeme der Technischen Universität Chemnitz durchgeführten Teilprojekte eingegangen. Die Umsetzung der entwickelten Ansätze und Werkzeuge wird anhand eines breiten Spektrums potentieller Einsatzmöglichkeiten im Web Engineering dargelegt.

Kapitel 8 schließt die Arbeit mit einer Zusammenfassung der Ergebnisse sowie einem Ausblick ab. Mehrere Anhänge ergänzen die Arbeit mit zusätzlichen Informationen zu den vorgestellten Ansätzen, Methoden und Werkzeugen.

2 Grundlagen und Prinzipien

Inhalt der folgenden Kapitel ist die Erarbeitung konstruktiver Maßnahmen im Rahmen der ganzheitlichen Betrachtung der Anwendungsentwicklung für Intelligente Umgebungen im Web Engineering. Aufgrund der wechselseitigen Beeinflussung realer Konstrukte, deren Repräsentation als auch Manipulation im World Wide Web, bedarf der geforderte Entwicklungsprozess einer gesonderten Betrachtung. Gegenstand dieses Kapitels ist somit zunächst die Analyse hierfür benötigter Prinzipien und die Einführung technologischer Grundlagen, die als Ausgangspunkt für die Erarbeitung weiterer Maßnahmen im späteren Verlauf dieser Arbeit dienen.

2.1 Ausgangssituation

Als Ausgangspunkt zur Ermittlung von Prinzipien dient ein typisches Beispiel für Intelligente Umgebungen: die *Hausautomation* (Thestrup, et al., 2008). Hierbei handelt es sich im weiteren Rahmen um die Integration von Gebäudetechnik, Kommunikationstechnologien und Kontrollelementen zur Gestaltung von Gebäuden und deren Infrastruktur, um eine einfache und komfortable Nutzung sowohl für den Betreiber als auch den Bewohner zu gewährleisten (Hydra Consortium, 2007).

Eine grundlegende Komponente des Gesamtsystems stellt zunächst die eingesetzte Geräteinfrastruktur dar. Hier können verschiedene konkurrierende Industriestandards, die auf unterschiedlichen Kommunikationsstandards und -technologien basieren, Einsatz finden (vgl. Abbildung 2). Hierbei ist es durchaus möglich, dass mehrere Infrastrukturen für verschiedene Anwendungsdomänen gleichzeitig Verwendung finden. So kann beispielsweise eine Infrastruktur zur Steuerung von Gebäudefunktionen (z.B. Beleuchtung, Schließmechanismen und Klimasteuerung) und eine zweite zur Kontrolle der im Haus befindlichen Unterhaltungselektronik eingesetzt werden. Weitere Infrastrukturen können zur Überwachung des Stromverbrauches oder zur Ermittlung von Umgebungs- und Wetterdaten (z.B. Temperatur, Luftfeuchtigkeit, Sonneneinstrahlung aber auch der Anwesenheit von Personen) eingesetzt werden.

Abhängig von den verwendeten Technologien, existieren unterschiedlichste Programmiermodelle. Diese reichen von proprietärer Software zur Konfiguration und Steuerung der Infrastruktur bis hin zu offenen Schnittstellen, beispielsweise