

Entwicklung großer Systeme mit generischen Methoden

Finanzierungsantrag

2001 - 2002 - 2003

Sonderforschungsbereich 501

Universität Kaiserslautern

April 2000

An die
Deutsche Forschungsgemeinschaft
- Sonderforschungsbereiche -
53170 Bonn

Antrag auf Finanzierung des Sonderforschungsbereichs 501

"Entwicklung großer Systeme mit generischen Methoden"

gefördert seit 1995
für die Jahre

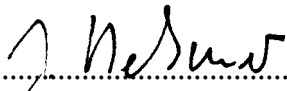
2001 - 2002 - 2003

Sprecherhochschule:
Universität Kaiserslautern

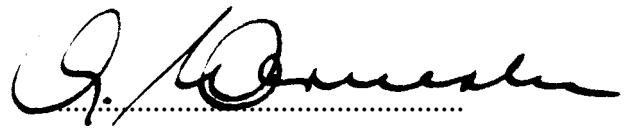
Sprecher:
Prof. Dr. Jürgen Nehmer
Sonderforschungsbereich 501
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Telefon: 0631 / 205-4020
Telefax: 0631 / 205-3289

Sekretariat des SFB:
Helena Alt
Sonderforschungsbereich 501
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Telefon: 0631 / 205-4025
Telefax: 0631 / 205-3289

Kaiserslautern, 1.2.2000


.....
Prof. Dr. J. Nehmer
(Sprecher)

Kaiserslautern, 1.2.2000


.....
Prof. Dr. G. Warnecke
(Präsident)



Inhaltsverzeichnis

0	Erklärung zur Raumsituation des SFB.....	ix
1	Allgemeine Angaben zum SFB	1
1.1	Forschungsprogramm	2
1.2	Übersicht über die Teilprojekte	5
1.2.1	Übersicht über die Teilprojekte, gegliedert nach Projektbereichen	5
1.3	Inhaltlich-strukturelles Umfeld und personelle Situation des SFB.....	6
1.4	Förderung des wissenschaftlichen Nachwuchses	6
1.4.1	Verzeichnis der laufenden Dissertationen	7
1.4.2	Besondere Maßnahmen zur Förderung des wissenschaftlichen Nachwuchses.....	9
1.5	Alphabetische Liste der am Sonderforschungsbereich beteiligten Wissenschaftler	10
1.6	Verzeichnis der Forschungsthemen, an denen Wissenschaftler des Sonderforschungsbereichs mitarbeiten und die in anderen Verfahren der Deutschen Forschungsgemeinschaft oder anderer Zuwendungsgeber gefördert werden oder für die Förderungsanträge gestellt sind	14
1.7	In der Ordnung des SFB vorgesehene Gremien und ihre personelle Zusammensetzung.....	15
2	Überblick über die bisherige und die beantragte Förderung des Sonderforschungsbereichs	16
2.1.1	Zusammenstellung der dem Sonderforschungsbereich bis einschließlich 2000 aus der Grundausrüstung zur Verfügung gestellten Sachmittel sowie des entsprechenden Ansatzes für 2001.....	16
2.1.2	Zusammenstellung der als Ergänzungsausstattung für den Sonderforschungsbereich bis einschließlich 2000 bewilligten und für die Haushaltsjahre 2001 bis 2003 beantragten Mittel.....	17
2.2	Anzahl der Mitarbeiter/innen, die aus der Grundausrüstung im HH-Jahr 2001 für die jetzt beantragten Teilprojekte zur Verfügung stehen werden.....	18

2.3	Anzahl der Mitarbeiter/innen, für die Personalmittel aus der Ergänzungsausstattung beantragt werden.....	19
2.3.1	für das Haushaltsjahr 2001	19
2.3.2	für das Haushaltsjahr 2002	20
2.3.3	für das Haushaltsjahr 2003	21
2.4	Als Ergänzungsausstattung beantragte Sondermittel für Sächliche Verwaltungsausgaben	22
2.4.1	für das Haushaltsjahr 2001	22
2.4.2	für das Haushaltsjahr 2002	23
2.4.3	für das Haushaltsjahr 2003	24
2.5	Liste der Geräte über DM 20.000,-- , die aus Mitteln für Sonderforschungsbereiche oder aus Mitteln der Grundausrüstung (HBFG, Mittel der Hochschule, des Landes) beschafft worden sind bzw. als Leihgabe der DFG zur Verfügung stehen oder deren Beschaffung 2000 aus Mitteln des SFB oder anderer Förderungsverfahren vorgesehen ist.....	25
2.6	Liste der als Ergänzungsausstattung für die HH-Jahre 2001 / 2002 / 2003 beantragten Geräte über DM 20.000,--	26
3	Darstellung des Programms nach Projektbereichen und Teilprojekten	27
	Projektbereich Z: "Anträge von allgemeinem Interesse"	27
	Projektbereich A: "Software-Entwicklungsumgebung"	35
	Teilprojekt A1.....	39
	Teilprojekt A2.....	63
	Teilprojekt A3.....	85
	Projektbereich B: "Generische Modelle und Methoden der Systementwicklung"	107
	Teilprojekt B1	111
	Teilprojekt B2.....	135
	Teilprojekt B4.....	157
	Teilprojekt B5.....	179
	Teilprojekt B10.....	197
	Teilprojekt B11	215
	Teilprojekt B12.....	243

Projektbereich C:"Beschreibungstechniken"	261
Teilprojekt C1	265
Projektbereich D:"Prototypanwendungen"	285
Teilprojekt D1	289
Teilprojekt D2	317
4 Geschäftsordnung des SFB	345

Erklärung zur Raumsituation des Sonderforschungsbereichs

ja nein

Decken die vorhandenen Räume den Bedarf des
Sonderforschungsbereichs zum Antragszeitpunkt? (x)

Wird der Raumbedarf in den Haushaltsjahren	2001	(x)
	2002	(x)
	2003	(x)

einschließlich einer etwa beantragten Erweiterung des
Sonderforschungsbereichs gedeckt werden?

Falls nein:

Wie sollen die benötigten Räume beschafft werden?

() durch Umverteilen vorhandener Räume

() durch Neu- bzw. Umbau

() durch Anmietung

Wie soll die vorstehend genannte Maßnahme finanziert werden?

() aus Haushaltsmitteln der Hochschule

() aus sonstigen der Hochschule zur Verfügung stehenden Mitteln

Beziehen Arbeitsgruppen des Sonderforschungsbereichs
im laufenden Jahr oder im Antragszeitraum neuerbaute
oder umgebaute Räume? (x)

Wenn ja, bitte den Zeitpunkt angeben:

Kaiserslautern, 1.2.2000

G. J. Conner

Der Präsident

(Prof. Dr. G. Warnecke)

Allgemeine Angaben zum Sonderforschungsbereich

1.1 Forschungsprogramm

Der SFB hat sich die Aufgabe gestellt, Verfahren zur Entwicklung großer Softwaresysteme, nachfolgend Systeme genannt, auf der Basis "generischer Methoden" zu erarbeiten. Unter dem Begriff "generische Methoden" fassen wir alle Beschreibungs- und Generierungstechniken mit den dazugehörigen Werkzeugen zusammen, durch die eine Wiederverwendung von existierenden Softwareprodukten, Entwicklungsschritten und Erfahrungen bei der Entwicklung eines neuen Systems methodisch unterstützt wird. In wohlverstandenen Teilbereichen soll diese methodische Unterstützung bis zur automatischen Softwaregenerierung getrieben werden. Wir erhoffen uns von den Ergebnissen des SFB wesentliche Impulse für eine ingenieurmäßige Softwaretechnologie, durch die - in Anlehnung an die klassische Konstruktionslehre des Maschinenbaus - große Systeme mit vorgegebenen Qualitätsanforderungen und abschätzbarem Personal- und Zeitaufwand erstellt werden können.

Das Forschungsprogramm des SFB ist in vier Projektbereiche gegliedert:

Im Projektbereich A wird ein Softwareentwicklungslabor betrieben, in das alle methodischen Ergebnisse, die Werkzeugreife erreicht haben, integriert werden. Es dient dazu, kontrollierte Entwicklungsprojekte in ausgewählten Anwendungsfeldern zu unterstützen. Die Architektur des Labors folgt dem "experience factory"-Ansatz von Basili.

Im Projektbereich B werden generische Produkt-, Prozess- und Qualitätsmodelle und -methoden entwickelt, in Experimenten erprobt und Werkzeuge zu ihrer leichten Handhabbarkeit implementiert.

Im Projektbereich C werden Beschreibungstechniken untersucht und ggf. erweitert, die für die unterschiedlichen Phasen im Software-Entwicklungsprozess benötigt werden. Ein besonderes Anliegen ist die Verfolgbarkeit aller Entwicklungsschritte, die eine Durchgängigkeit zwischen verschiedenen formalen und informellen Beschreibungstechniken verlangt. Der Projektbereich C unterstützt außerdem die anderen Projektbereiche bei der Auswahl geeigneter Beschreibungstechniken.

Im Projektbereich D wird das Expertenwissen auf ausgewählten Anwendungsfeldern gesammelt, strukturiert und zur Wiederverwendung in Softwareentwicklungsprojekten aufbereitet (domain engineering). Aufgabe des Projektbereichs ist es auch, Testumgebungen wie Hardwareplattformen und Simulatoren für prototypische Anwendungen in einem ausgewählten Anwendungsfeld bereitzustellen. Domänenwissen und Unterstützungssysteme werden selbst unter dem Aspekt der Wiederverwendung strukturiert.

Im Projektbereich Z wird der SFB verwaltet.

Eine ausführliche Übersicht über das Forschungsprogramm des SFB enthält der Erstantrag.

In der ersten Förderperiode hat sich die überwiegende Zahl der Projekte schwerpunktmäßig mit unterschiedlichen generischen Ansätzen auf dem Gebiet der Produkt- und Prozessmodellierung beschäftigt.

Die zweite Förderperiode war durch Anstrengungen gekennzeichnet, die in den Teilprojekten erarbeiteten Konzepte weiterzuentwickeln, die Ergebnisse aufeinander abzustimmen und zu integrieren, um so eine Durchgängigkeit von Teilprozessen in der Unterstützung einer Systementwicklung zu erreichen. Desweiteren gingen von dem in der zweiten Förderperiode neu aufgenommenen Teilprojekt A3 wichtige Impulse für eine Strukturierung der Erfahrungsdatenbank inkl. der notwendigen Zugriffsverfahren aus. Eine erste Version der Erfahrungsdatenbank wurde in enger Kooperation von A3 mit A1/B1/B2 aufgrund durchgeführter prototypischer Entwicklungsprojekte in dem Anwendungsfeld Gebäudeautomation erprobt.

Eine von Beginn des SFB an praktizierte neue Form der interdisziplinären Zusammenarbeit unter den Teilprojekten hat sich sehr bewährt und ist auch in der zweiten Förderperiode fortgesetzt worden: die projektübergreifende Durchführung von Aufgaben in Querschnittprojekten.

In der zweiten Förderperiode wurden folgende Querschnittprojekte durchgeführt:

Q1) Prototyping eines Gebäudeautomationssystems

In diesem Querschnittprojekt wurden zwei aufeinander aufbauende kontrollierte Entwicklungsprojekte durchgeführt, deren Ziel die Erstellung ausführbarer Anforderungen in SDL war.

Q2) Entwurf eines Gebäudeautomationssystems

In diesem Querschnittprojekt wurde ein Entwicklungsprojekt mit Studenten durchgeführt, das die Phasen Entwurf und Codierung einschloss.

Q3) Erarbeitung eines Domänenmodells

Dieses Querschnittprojekt hat die Aufgabe, unter Rückgriff auf etablierte Methoden des Domain-Engineering das vorhandene Wissen über das Anwendungsfeld Gebäudeautomation zu strukturieren und als Basis einer systematischen Wiederverwendung in ein Domänenmodell zu gießen.

Q4) Anforderungsanalyse eines Gebäudeautomationssystems

Im Rahmen dieses Querschnittprojektes wurden aufbauend auf den Ergebnissen der Kooperation der Teilprojekte B4 und C1 mehrere natürlich-sprachliche Problembeschreibungen aus dem Anwendungsfeld Gebäudeautomation in eine formale, objektorientierte Problemspezifikation überführt. Die Ziele dabei waren zum einen, allgemein die Eignung des entwickelten Ansatzes für dieses Anwendungsfeld zu überprüfen

fen, und zum anderen, speziell die Möglichkeiten der Wiederverwendung von Teilen einer Problemspezifikation zu untersuchen.

Die Mitarbeiter der Teilprojekte tragen bis zu 8 Std./Woche zu den Querschnittsprojekten bei.

Die Arbeiten und Ergebnisse der Querschnittprojekte erscheinen im Ergebnisbericht als eigenständige Berichte.

1.2 Übersicht über Teilprojekte

1.2.1 Übersicht über die Teilprojekte, gegliedert nach Projektbereichen

Kenn-ziffer	Titel	Fachgebiet und Arbeitsrichtung	Leiter des Teilprojekts
A1	SE-Labor	Software-Engineering	Prof. Rombach
A2	Entwicklung einer flexiblen Modellierungs- und Ausführungsumgebung für Software-Entwicklungsprozesse	Software-Engineering	Prof. Richter / Prof. Rombach
A3	Unterstützung des Software-Entwicklungsprozesses durch objekt-relationale Datenbank-technologie	Datenbanksysteme	Prof. Härder / Dr. Ritter
B1	Generische Modellierung von Prozessen und Experimenten	Software-Engineering	Prof. Rombach
B2	Flexible Planung und Steuerung von Software-Entwicklungsprozessen	Software-Engineering	Prof. Richter
B4	Generische Kommunikationssysteme	Kommunikationssysteme	Prof. Gotzhein
B5	GeneSys - Generische Systemsoftware	Betriebssysteme	Prof. Nehmer
B10	Anwendungsentwicklung mit vorkonfektionierten Systemen	Software- Engineering	Prof. Nehmer
B11	Zuverlässige eingebettete Systeme	Software-Engineering	Dr. Schürmann
B12	Stratified Component Frameworks	Software-Engineering	Prof. Atkinson / Dr. Kühne
C1	Formale Beschreibungstechniken	Software-Spezifikation	Prof. Avenhaus / Prof. Madlener
D1	Anwendungssystem Gebäude	Anwendungsfeld Gebäude-automation	Prof. Zimmermann
D2	Modellbasierte Entwicklung wiederverwendbarer Regelungsalgorithmen	Anwendungsfeld Gebäude-automation	Prof. Litz

1.2.2 Übersicht über die im letzten Bewilligungszeitraum geförderten Teilprojekte, deren Fortführung nicht beabsichtigt ist

entfällt

1.3 Inhaltlich-strukturelles Umfeld und personelle Situation des Sonderforschungsbereichs

Der Sonderforschungsbereich ist schwerpunktmäßig im Fachbereich Informatik der Universität Kaiserslautern angesiedelt. Mit gegenwärtig 18 Professuren und einer stark ingenieurwissenschaftlichen Ausrichtung der vorhandenen Studiengänge "Informatik", "Technoinformatik" und "Angewandte Informatik" bietet die Universität gute Voraussetzungen für eine erfolgreiche Bearbeitung des SFB-Themas. Durch zwei angelagerte Institute - das Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI) und das 1996 neu gegründete Fraunhofer-Institut für Experimentelles Software-Engineering (IESE) bestehen ideale Voraussetzungen, Ergebnisse der Grundlagenforschung in praxisnahen Projekten zu erproben. Die gute Zusammenarbeit zwischen dem SFB und diesen Instituten wird durch die starke personelle Verflechtung garantiert (Prof. Rombach, der Leiter des IESE, ist Mitglied des Fachbereichs und Vorstandsmitglied im SFB).

Prof. Litz vom Fachbereich Elektro- und Informationstechnik ist Experte in der Anwendungsdomäne "Gebäudeautomation". Seine Mitarbeit im SFB 501 halten wir für unverzichtbar, um Automatisierungstechnische Fragestellungen und SE-Ansätze sinnvoll zu verknüpfen.

In der zurückliegenden Förderperiode wurden zwei C3-Stellen (Prof. Atkinson, Prof. Henrich) und eine C2-Stelle (Dr. Schürmann) mit einem unmittelbaren thematischen Bezug zum Arbeitsgebiet des SFB neu besetzt. Alle neuen Kollegen haben Interesse an einer Mitarbeit im SFB geäußert und teilweise bereits Anträge für die kommende Förderperiode gestellt. Das Engagement des IESE im SFB wird von uns besonders begrüßt. Es verstärkt nicht nur die Personalkapazität des SFB, sondern hilft auch, Erkenntnisse der Grundlagenforschung schneller in die Praxis zu bringen.

In der dritten Förderperiode werden die Kontakte einzelner Teilprojekte zur Industrie mit dem Ziel verstärkt, Transferprojekte aufzusetzen.

1.4 Förderung des wissenschaftlichen Nachwuchses

Die Förderung des wissenschaftlichen Nachwuchses geschieht durch

- Anleitung nicht-promovierter Mitarbeiter zum wissenschaftlichen Arbeiten,
- Einbeziehung erfahrener promovierter wissenschaftlicher Mitarbeiter in die Leitung von Teilprojekten,

- eine SFB-Seminarreihe, in der Wissenschaftler des SFB regelmäßig über ihre Forschungsarbeiten berichten,
- Einladung externer Wissenschaftler zu Vorträgen im Rahmen des SFB-Seminars.

1.4.1 Verzeichnis der laufenden Dissertationen

Baum, Lothar:	"Generating customized run-time platforms based on application requirements"
Becker, Martin:	"Entwicklung generativer Betriebssysteme"
Ciolkowski, Marcus:	"Generische Methodik für SE-Experimente"
Diehl, Andreas:	"Neue Automatisierungsstrategien zur Optimierung des Kälteerzeugungsprozesses"
Denu, Patrick:	"Robuste modellprädiktive Regler"
Eschbach, Robert:	"A Refinement Calculus for Abstract State Machines"
Feldmann, Raimund:	"Generische Konzepte zur Erstellung von Wiederverwendungs-Repositories"
Geppert, Birgit:	"The SDL-Pattern Approach - A Reuse-Driven SDL Methodology for Designing Communication Software Systems"
Geyer, Lars:	"A tool-based approach for the mapping of application requirements to a software architecture"
Goldmann, Sigrid:	"Knowledge-Based Planning and Replanning Support in Software"
Holz, Harald:	"Process-Based Knowledge Management Support for Software Engineering"
Kötting, Boris:	"Agentenbasierte Unterstützung von Planungsprozessen in virtuellen Unternehmen"
Kronenburg, Martin:	"Ein Ansatz zur Erstellung präziser, verständlicher Problemspezifikation großer reaktiver Systeme"
Mahnke, Wolfgang:	"Versionierungskonzepte für SE-Repositories"
Marder, Ulrich:	"Medienspezifische Datentypen für objekt-relationale DBMS"

Metzger, Andreas:	"Entwurfsbegleitendes Prototyping"
Merz, Rolf:	"Effizienz generischer Modelle in der mathematischen Modellbildung"
Molter, Georg:	"The Core Systems Concept - Integrating Architecture-based and Reuse-Aware Software Development"
Münch, Jürgen:	"Zielführende, wiederverwendungsorientierte Erstellung von Software-Entwicklungsvorgehensmodellen"
Peper, Christian:	"Eigenschaftsorientierte Spezifikation und korrektkeitserhaltende Komposition generischer Echtzeit-Kommunikationsdienste"
Queins, Stefan:	"Entwurf einer modell- und generatorbasierten Software-Entwicklungsumgebung"
Riegel, Jan-Peter:	"Domänenspezifische Modellierung in den frühen Phasen des Software-Entwurfs"
Rössler, Frank:	"Scenario and Reuse Driven Design of Distributed Systems based on SDL"
Schaaf, Martin:	"Intelligent Process Driven Knowledge Acquisition"
Schaible, Philipp:	"Engineering of Application-Tailored Communication Middleware"
Steiert, Hans-Peter:	"Ein objektorientierter, generativer Ansatz zum Einsatz von ORDBMS in Entwurfsumgebungen"
Tetteroo, Torsten:	"Entwurf zuverlässiger eingebetteter Systeme"
Thees, Joachim:	"Automatische Generierung effizienter Protokollimplementierungen mit FDTs"
von Knethen, Antje:	"Change-oriented requirements traceability: Support for the evolution of embedded software"
Voos, Holger:	"Market-based Control: Eine neue Methode der Automatisierungstechnik"
Vorwieger, Stefan:	"Umfassende wiederverwendungsorientierte Verwaltung von SE-Erfahrung"
Weng, Xiying:	"Verifikation industrieller Steuerungsalgorithmen"

Zhang, Nan: "Supporting Semantically Rich Relationships in Extensible Object-Relational Database Management Systems"

Zhang, Weiping: "Beurteilung des Leistungsverhaltens objekt-relationaler DBMS"

1.4.2 Besondere Maßnahmen zur Förderung des wissenschaftlichen Nachwuchses

entfällt

1.5 Alphabetische Liste der am Sonderforschungsbereich beteiligten Wissenschaftler

(Mitglieder im Sinne der Ordnung des Sonderforschungsbereichs sind mit * vor dem Namen gekennzeichnet)

Grundausrüstung = GA / Ergänzungsausrüstung = EA

Name, Vorname, akad. Grad, Dienststellung	Fachrichtung	Fachbereich/ Einrichtung	GA/ EA	Teil- projekt
*Atkinson, Colin Prof. Dr.	Informatik	FB Informatik	GA	B12
*Avenhaus, Jürgen Prof. Dr.	Informatik	FB Informatik	GA	C1
Baum, Lothar Dipl.-Inform.	Informatik	FB Informatik	EA	B5
Becker, Martin Dipl.-Inform.	Informatik	FB Informatik	GA	B10
Ciolkowski, Marcus Dipl.-Inform.	Informatik	FB Informatik	GA	B1
Denu, Patrick Dipl.-Ing.	E+I-Technik	FB E+I-Technik	GA	D2
Diehl, Andreas Dipl.-Ing.	E+I-Technik	FB E+I-Technik	GA	D2
Eschbach, Robert Dipl.-Inform.	Informatik	FB Informatik	GA	C1
Feldmann, Raimund Dipl.-Inform.	Informatik	FB Informatik	EA	A1
Gilbert, Andreas Dipl.-Inform.	Informatik	FB Informatik	GA	B10
Goldmann, Sigrid Dipl.-Inform.	Informatik	FB Informatik	EA	A2
*Härder, Theo Prof. Dr.	Informatik	FB Informatik	GA	A3

*Gotzhein, Reinhard Prof. Dr.	Informatik	FB Informatik	GA B4
Holz, Harald Dipl.-Inform.	Informatik	FB Informatik	EA B2
Koetting, Boris Dipl.-Inform.	Informatik	FB Informatik	GA B2
Krechel, Dirk Dipl.-Inform.	Informatik	FB Informatik	GA A2
Kronenburg, Martin Dipl.-Inform.	Informatik	FB Informatik	EA C1
*Dr. Kühne, Thomas Dipl.-Inform.	Informatik	FB Informatik	GA B12
*Litz, Lothar Prof. Dr.	E+I-Technik	FB E+I-Technik	GA D2
*Madlener, Klaus Prof. Dr.	Informatik	FB Informatik	GA C1
Mahnke, Wolfgang Dipl.-Inform.	Informatik	FB Informatik	EA A3
Marder, Ulrich Dipl.-Inform.	Informatik	FB Informatik	EA A3
Metzger, Andreas Dipl.-Inform.	Informatik	FB Informatik	GA D1
Dr. Merz, Rolf Dipl.-Phys.	E+I-Technik	FB E+I-Technik	EA D2
Molter, Georg Dipl.-Inform.	Informatik	FB Informatik	GA B10
Münch, Jürgen Dipl.-Inform.	Informatik	FB Informatik	EA B1
*Nehmer, Jürgen Prof. Dr.	Informatik	FB Informatik	GA B5/B10
Peper, Christian Dipl.-Inform.	Informatik	FB Informatik	EA B4

Petersen, Erik Dipl.-Inform.	Informatik	FB Informatik	EA	A2
Queins, Stefan Dipl.-Inform.	Informatik	FB Informatik	EA	D1
*Richter, Michael Prof. Dr.	Informatik	FB Informatik	GA	A2/B2
Riegel, Jan-Peter Dipl.-Inform.	Informatik	FB Informatik	EA	D1
*Dr. Ritter, Norbert Dipl.-Inform.	Informatik	FB Informatik	GA	A3
*Rombach, Dieter Prof. Dr.	Informatik	FB Informatik	GA	A1/B1
Schaaf, Martin Dipl.-Inform.	Informatik	FB Informatik	EA	B2
Schaible, Philipp Dipl.-Inform.	Informatik	FB Informatik	GA	B4
Schmitt, Sascha Dipl.-Inform.	Informatik	FB Informatik	GA	B2
*Dr. Schürmann, Bernd Dipl.-Inform.	Informatik	FB Informatik	GA	B11
Steiert, Hans-Peter Dipl.-Inform.	Informatik	FB Informatik	GA	A3
Tetteroo, Torsten Dipl.-Inform.	Informatik	FB Informatik	GA	B11
Thees, Joachim Dipl.-Inform.	Informatik	FB Informatik	GA	B4
von Knethen, Antje Dipl.-Inform.	Informatik	FB Informatik	GA	B1
Voos, Holger Dipl.-Ing.	E+I-Technik	FB E+I-Technik	GA	D2
Vorwieger, Stefan Dipl.-Inform.	Informatik	FB Informatik	EA	A1

Weng, Xiyang M.Sci.	E+I-Technik	FB E+I-Technik	GA	D2
Zhang, Nan M.Sc.	Informatik	FB Informatik	GA	A3
Zhang, Weiping Dipl.-Ing.	Informatik	FB Informatik	GA	A3
*Zimmermann, Gerhard Prof. Dr.	Informatik	FB Informatik	GA	D1

1.6 Verzeichnis der Forschungsthemen, an denen Wissenschaftler des Sonderforschungsbereichs mitwirken und die in anderen Verfahren der Deutschen Forschungsgemeinschaft oder anderer Zuwendungsgeber gefördert werden oder für die Förderungsanträge gestellt worden sind

Thema	Name der beteiligten Wissenschaftler	Zuwendungsgeber und Aktenzeichen oder Antragsdatum	Kennziffer des im SFB beantragten Teilprojekts
WEBSELL	Prof. Richter, Schmitt	EU-Projekt Contract-No. 27068	A2, B2
WEBPLAN	Prof. Richter	DFG, RI327/14-1	A2, B2
CYCLOPS	Prof. Richter	BMBF (Reisemittel) BRA-INF-CYCLOPS	A2, B2
FIORES	Prof. Richter	EU-Projekt	A2, B2
Forschungsstelle Mathematische Logik	Prof. Richter	Heidelberger Akademie der Wissenschaften	A2, B2
MILOS	Prof. Richter, Goldmann, Holz, Kötting, Schaaf	BMBF (Reisemittel) KAN 99/007 INF	A2, B2
Rechnergestütztes Informationssystem für technische Anwendungen	Prof. Härder	Stiftung für Innovation Rheinland-Pfalz	A3
Middleware-Technologie	Prof. Härder	Daimler-Chrysler, Ulm	A3
Entwurf und Analyse industrieller Steuerungsalgorithmen unter Berücksichtigung internationaler Industriestandards	Prof. Litz	Stiftung für Innovation Rheinland-Pfalz	D2

1.7 In der Ordnung des Sonderforschungsbereichs vorgesehene Gremien (außer Mitgliederversammlung) und ihre personelle Zusammensetzung

Sprecher: Prof. Dr. Jürgen Nehmer

stellv. Sprecher: Prof. Dr. Dieter Rombach

Vorstand: Prof. Dr. Colin Atkinson

Prof. Dr. Jürgen Avenhaus

Prof. Dr. Reinhard Gotzhein

Prof. Dr. Theo Härder

Dr. Thomas Kühne

Prof. Dr. Lothar Litz

Prof. Dr. Klaus Madlener

Prof. Dr. Jürgen Nehmer

Prof. Dr. Michael Richter

Dr. Norbert Ritter

Prof. Dr. Dieter Rombach

Dr. Bernd Schürmann

Prof. Dr. Gerhard Zimmermann

2 Überblick über die bisherige und die beantragte Förderung des Sonderforschungsbereichs

2.1.1 Zusammenstellung der dem Sonderforschungsbereich aus der Grundausrüstung im letzten Förderungszeitraum zur Verfügung gestellten Sachmittel sowie des entsprechenden Ansatzes für 2001

SV: Sächliche Verwaltungsausgaben - I: Investitionsmittel (Geräte über 20.000,--)

(alle Angaben in TDM)

Haushalts- jahr	GA Hochschule			GA beteiligte Einrichtungen			GA Sachmittel insgesamt
	SV	I	Summe	SV	I	Summe	
1998	95.000	120.000	215.000	-.-	-.-	-.-	215.000
1999	25.000	115.000	140.000	-.-	-.-	-.-	140.000
2000	25.000	115.000	140.000	-.-	-.-	-.-	140.000
2001	30.000	-.-	30.000				30.000

2.1.2 Zusammenstellung der als Ergänzungsausstattung für den Sonderforschungsbereich

- bis einschließlich 2000 bewilligten
- für die Haushaltsjahre 2001 bis 2003 beantragten Mittel

PK: Personalkosten

SV: Sächliche Verwaltungsausgaben

I: Investitionen (Geräte über 20.000,--)

Haushalts-jahr	Ergänzungsausstattung			gesamt
	PK	SV	I	
bis 1997	4.856.000	301.500	219.400	5.376.900
1998	2.155.200	204.500	.-	2.359.700
1999	2.199.600	203.500	.-	2.403.100
2000	2.134.800	191.900	.-	2.326.700
Zwischensumme				12.466.400
2001	3.196.800	233.000	150.000	3.579.800
2002	3.196.800	226.000	65.000	3.487.800
2003	3.196.800	218.000	65.000	3.479.800

2.2 Anzahl der Mitarbeiter/innen, die aus der Grundausstattung 2000 für die jetzt beantragten Teilprojekte zur Verfügung stehen werden

Besoldungsgruppe / Vergütungsgruppe	<u>Anzahl</u> der beteiligten Mitarbeiter/innen (ohne beratende Wissenschaftler)
Wissenschaftliches Personal	
C4 / Sondervertrag	6
C3 / A16	4
C2 / A15	1
C1 / A14	1
A13	
BAT I	
BAT Ia	
BAT Ib	
BAT IIa	18
Summe wiss. Personal	30
wissenschaftliche Hilfskräfte	2
studentische Hilfskräfte	
Nichtwissenschaftliches Personal	
BAT III	
BAT IVa	1
BAT Va/b	1
BAT Vc	
BAT VIb	
BAT VII -X/MTL	
Summe nichtwissensch. Personal	2

2.3 Anzahl der Stellen, für die Personalmittel als ERGÄNZUNGS-AUSSTATTUNG beantragt werden

2.3.1 Haushaltsjahr 2001 (in Klammern ist die bisher bewilligte Anzahl angegeben)

Teil-projekt	Son-der-vertrag	I-Ib	Ila	Ila/2	Sum-me Sond.-BAT Ila/2	wiss. Hilfs-kräfte	stud. Hilfs-kräfte	BAT III	BAT IVa	BAT IVb	BAT Va/b	BAT Vc	BAT VIa	BAT VIb	BAT VIII-X, MTL	Sum-me BAT III-X u. MTL	insge-samt (ohne Hilfs-kräfte)
A1			2 (2)				7 (-)										2
A2			2 (1)				3 (-)										2
A3			2 (2)				3 (-)										2
B1			2 (1)				2 (-)										2
B2			2 (2)				2 (-)										2
B4			2 (2)				2 (-)										2
B5			2 (2)				2 (-)										2
B10			2 (1)				2 (-)										2
B11			1 (-)				1 (-)										1
B12			1 (-)				2 (-)										1
C1			2 (1)				2 (-)										2
D1			2 (2)				2 (-)										2
D2			2 (2)				2 (-)										2
Z							- (14)				1 (1)						1
ins-ges.			24 (18)				32 (14)				1 (1)						25(19)
+/-			+ 6				+ 18				-						+ 6

2.3.2 Haushaltsjahr 2002

Teil- pro- jekt	Son- der- ver- trag	I-Ib	IIa	IIa/2	Sum- me Sond. -BAT IIa/2	wiss. Hilfs- kräfte	stud. Hilfs- kräfte	BAT III	BAT IVa	BAT IVb	BAT Va/b	BAT Vc	BAT VIa	BAT VIb	BAT VIII- X, MTL	Sum- me BAT III-X u. MTL	insge- samt (ohne Hilfs- kräfte)
A1			2				7										2
A2			2				3										2
A3			2				3										2
B1			2				2										2
B2			2				2										2
B4			2				2										2
B5			2				2										2
B10			2				2										2
B11			1				1										1
B12			1				2										1
C1			2				2										2
D1			2				2										2
D2			2				2										2
Z			-				-				1						1
ins- ges.			24				32				1						25
+/-			-				-				-						-

2.3.3 Haushaltsjahr 2003

Teil- pro- jekt	Son- der- ver- trag	I-Ib	IIa	IIa/2	Sum- me Sond. -BAT IIa/2	wiss. Hilfs- kräfte	stud. Hilfs- kräfte	BAT III	BAT IVa	BAT IVb	BAT Va/b	BAT Vc	BAT VIa	BAT VIb	BAT VIII- X, MTL	Sum- me BAT III-X u. MTL	insge- samt (ohne Hilfs- kräfte)
A1			2				7										2
A2			2				3										2
A3			2				3										2
B1			2				2										2
B2			2				2										2
B4			2				2										2
B5			2				2										2
B10			2				2										2
B11			1				1										1
B12			1				2										1
C1			2				2										2
D1			2				2										2
D2			2				2										2
Z			-				-				1						1
ins- ges.			24				32				1						25
+/-			-				-				-						-

2.4 Als ERGÄNZUNGS-AUSSTATTUNG beantragte Mittel für Sächliche Verwaltungsausgaben

2.4.1 Haushaltsjahr 2001

Teilprojekt	Kleingeräte bis DM 20.000	Verbrauchs- mittel	Reisemittel	Bücher und Zeitschriften	Druckkosten	Vervielfältigungen	Sonstiges (Lizenzen, Gast. u. Kol- loqu)	gesamt
A1	-	15.000	-	-	-	-	-	15.000
A2	-	3.000	-	-	-	-	-	3.000
A3	-	3.000	-	-	-	-	-	3.000
B1	-	3.000	-	-	-	-	-	3.000
B2	-	3.000	-	-	-	-	-	3.000
B4	-	3.000	-	-	-	-	-	3.000
B5	-	3.000	-	-	-	-	-	3.000
B10	-	3.000	-	-	-	-	-	3.000
B11	-	3.000	-	-	-	-	-	3.000
B12	-	5.000	-	-	-	-	-	5.000
C1	-	3.000	-	-	-	-	-	3.000
D1	-	20.000	-	-	-	-	-	20.000
D2	-	19.000	-	-	-	-	-	19.000
Z	-	2.000	60.000	5.000	10.000	-	70.000	147.000
gesamt	-	88.000	60.000	5.000	10.000	-	70.000	233.000
+ / -	- 14.400	+ 57.500	+ 15.000	+ 5.000	- 2.000	-	- 20.000	+ 41.100

2.4.2 Haushaltsjahr 2002

Teilprojekt	Kleingeräte bis DM 20.000	Verbrauchs- mittel	Reisemittel	Bücher und Zeitschriften	Druckkosten	Vervielfältigungen	Sonstiges (Gastw. u. Kolloquia)	gesamt
A1	-	15.000	-	-	-	-	-	15.000
A2	-	3.000	-	-	-	-	-	3.000
A3	-	3.000	-	-	-	-	-	3.000
B1	-	3.000	-	-	-	-	-	3.000
B2	-	3.000	-	-	-	-	-	3.000
B4	-	3.000	-	-	-	-	-	3.000
B5	-	3.000	-	-	-	-	-	3.000
B10	-	3.000	-	-	-	-	-	3.000
B11	-	3.000	-	-	-	-	-	3.000
B12	-	3.000	-	-	-	-	-	3.000
C1	-	3.000	-	-	-	-	-	3.000
D1	-	15.000	-	-	-	-	-	15.000
D2	-	19.000	-	-	-	-	-	19.000
Z	-	2.000	60.000	5.000	10.000	-	70.000	147.000
gesamt	-	81.000	60.000	5.000	10.000	-	70.000	226.000
+ / -	-	- 7.000	-	-	-	-	-	- 7.000

2.4.3 Haushaltsjahr 2003

Teilprojekt	Kleingeräte bis DM 20.000	Verbrauchs- mittel	Reisemittel	Bücher und Zeitschriften	Druckkosten	Vervielfältigungen	Sonstiges (Gastw. u. Kolloquia)	gesamt
A1	-	15.000	-	-	-	-	-	15.000
A2	-	3.000	-	-	-	-	-	3.000
A3	-	3.000	-	-	-	-	-	3.000
B1	-	3.000	-	-	-	-	-	3.000
B2	-	3.000	-	-	-	-	-	3.000
B4	-	3.000	-	-	-	-	-	3.000
B5	-	3.000	-	-	-	-	-	3.000
B10	-	3.000	-	-	-	-	-	3.000
B11	-	3.000	-	-	-	-	-	3.000
B12	-	3.000	-	-	-	-	-	3.000
C1	-	3.000	-	-	-	-	-	3.000
D1	-	10.000	-	-	-	-	-	10.000
D2	-	12.000	-	-	-	-	-	12.000
Z	-	2.000	60.000	5.000	14.000	-	70.000	151.000
gesamt	-	69.000	60.000	5.000	14.000	-	70.000	218.000
+ / -	-	- 12.000	-	-	+ 4.000	-	-	- 8.000

2.5 Liste der Geräte über DM 20.000,--

- die aus Mitteln des Sonderforschungsbereichs oder aus Mitteln der Grundausstattung (HBFG, Mittel der Hochschule, des Landes) beschafft worden sind oder als Leihgabe der Deutschen Forschungsgemeinschaft zur Verfügung stehen oder
- deren Beschaffung 2000 aus Mitteln des Sonderforschungsbereichs oder anderer Förderungsverfahren beabsichtigt ist

Bezeichnung des Gerätes	Anschaffungsjahr	Anschaffungspreis	eingesetzt i.d. Teilprojekten	finanziert durch SFB/HBFG/Hochschule/Land, Leihgabe
SUN-Enterprise-Workstation mit Zubehör	1998	64.000	A3	Land
2 SUN-Enterprise-Workstation mit Zubehör	1999	115.000	A1	Land
Summe		179.000		

2.6 Liste der als Ergänzungsausstattung für die Haushaltsjahre 2001 / 2002/ 2003 beantragten Geräte über 20.000,-- (brutto)

Teil-projekt	Bezeichnung	Einzel-preis 2001	Einzel-preis 2002	Einzel-preis 2003
A1	Fortführende Lizenzen und Beschaffung neuer Werkzeuge (Gesamtpreise)	65.000	65.000	65.000
A1	Ersatz von Hardware	60.000	.-	.-
B11	Workstation mit X-Terminal	25.000		
	Summe	150.000	65.000	65.000

Projektbereich Z

Infrastruktur

3.1 Teilprojekt Z

3.1.1 Thema

Anträge von allgemeinem Interesse

3.1.2 Fachgebiet und Arbeitsrichtung

In diesem Projektbereich wird die übergreifende Infrastruktur für den gesamten SFB bereitgestellt. Dazu gehören die allgemeine Verwaltung, Reisemittel für SFB-Mitglieder, Gastwissenschaftler-Mittel sowie Mittel für die Organisation von Tagungen und Kolloquien.

Sächliche Verwaltungsaufgaben sind in diesem Projektbereich für teilprojektübergreifende Zwecke, wie z.B. Druck- und Kopierkosten für SFB-Berichte, zur Vorbereitung der Begehungen, Workshops etc. und allgemeine Literatur zur SFB-Thematik ausgewiesen.

3.1.3 Leiter

alle Teilprojektleiter

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ ja	☒ nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ ja	☒ nein
Tierversuche	☐ ja	☒ nein
gentechnologische Untersuchungen	☐ ja	☒ nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Jeweils Ergänzungsausstattung (Beträge in DM 1000.-).

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungs-ausgaben	Investitionen	Gesamt
bis 1997	210,0	199,0		409,0
1998	340,8	129,0	-	469,8
1999	342,0	129,0	-	471,0
2000	360,0	129,0	-	489,0
Zwischensumme	1252,8	586,0	-	1838,8
2001	74,4	147,0	-	221,4
2002	74,4	147,0	-	221,4
2003	74,4	151,0	-	225,4

3.2 Zusammenfassung

entfällt

3.3 Stand der Forschung

entfällt

3.4 Eigene Vorarbeiten

entfällt

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

entfällt

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

entfällt

3.7 Ergänzungsausstattung für das Teilprojekt

PK Bewilligung 2000		2001			2002			2003		
Vergütungs- gruppe	Anzahl	Vergütungs- gruppe	Anzahl	Betrag in DM	Vergütungs- gruppe	Anzahl	Betrag in DM	Vergütungs- gruppe	Anzahl	Betrag in DM
Va/b	1	Va/b	1	74.400	Va/b	1	74.400	Va/b	1	74.400
zusammen	1	zusammen	1	74.400	zusammen	1	74.400	zusammen	1	74.400
SV	Bezeichnung			Betrag in DM	Bezeichnung		Betrag in DM	Bezeichnung		Betrag in DM
	Verbrauchsmaterial			2.000	Verbrauchsmaterial		2.000	Verbrauchsmaterial		2.000
	Fachliteratur			5.000	Fachliteratur		5.000	Fachliteratur		5.000
	Druckkosten			10.000	Druckkosten		10.000	Druckkosten		14.000
	Reisemittel			60.000	Reisemittel		60.000	Reisemittel		60.000
	Gastwissenschaftler			50.000	Gastwissenschaftler		50.000	Gastwissenschaftler		50.000
	Kolloquia			20.000	Kolloquia		20.000	Kolloquia		20.000
	zusammen			147.000	zusammen		147.000	zusammen		151.000

3.7.1 Begründung des Personalbedarfs

	Name, Dienststellung	Bez. des Institutes der Hochschule	Anteil der aufgew. Gesamtarbeitszeit	im SFB tätig seit	derzeitige Einstufung
nichtwiss. Mitarbeiter	Alt, Helena	Univ. Kaiserslautern	38,5	1995	BAT Vb

Die Verwaltungsstelle BAT Vb wird für die anfallenden Verwaltungstätigkeiten im SFB benötigt. Dazu zählen u.a.:

- Durchführung aller Sachmittelbeschaffungen,
- Personalmittelverwaltung,
- Personaleinstellungen,
- Kontoführung und -überwachung,
- Koordination der Dienstreisen,
- Vorbereitung und Abrechnung von Gastwissenschaftler-Aufenthalten,
- Organisation von Seminaren und Kolloquien,
- Vorbereitung von Vorstands- und Mitgliedersitzungen,
- Anfertigung von Sitzungsprotokollen,
- Vorbereitung von Anträgen, Berichtskolloquien usw.
- Organisation der Begehungen.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben			
- stehen als Grundausrüstung voraussichtlich zur Verfügung	20.000	20.000	20.000
- werden als Ergänzungsausstattung beantragt			
Reisekosten	60.000	60.000	70.000
Druck- und Kopierkosten	10.000	10.000	14.000
Fachbücher und Zeitschriften	5.000	5.000	5.000
Verbrauchsmaterial	2.000	2.000	2.000
Gastwissenschaftler	50.000	50.000	50.000
Kolloquia	20.000	20.000	20.000
Summe (entspricht den Summen „sächliche Verwaltungsausgaben“ in Übersicht 3.7)	147.000	147.000	151.000

Die internationalen Kontakte des SFB - insbesondere zu Universitäten in den USA, Kanada und Australien - konnten in der zweiten Förderperiode deutlich verstärkt werden, so daß mit einem größeren Mittelbedarf in der dritten Förderperiode gerechnet wird.

Besonders enge Kontakte bestehen zu folgenden Wissenschaftlern:

- Prof. Parnas, Mc Masters University, Hamilton, Ontario /Canada
- Prof. Maurer, Universität Calgary / Canada (Wegberufung aus dem Teilprojekt B2 im Jahr 1998)
- Prof. Basili, Universität Maryland / USA
- Prof. Jeffery, New South Wales University, Sidney / Australien

Die Kontaktpflege mit international führenden Wissenschaftlern erachten wir als besonders wichtig. Hierfür beantragen wir 50.000 DM als Gastwissenschaftler-Mittel. Darüber hinaus beabsichtigen wir, in- und ausländische Wissenschaftler zu Vorträgen einzuladen, die im engeren Arbeitsgebiet des SFB liegen.

Zur Durchführung von Tagungsreisen und den Besuch in- und ausländischer Kollegen, mit denen der SFB kooperiert, werden jeweils 60.000 DM für die Haushaltsjahre 2001 - 2003 beantragt.

Für teilprojektübergreifende Druck- und Kopierkosten werden in den Haushaltsjahren 2001 und 2002 jeweils 10.000 DM, im Haushaltsjahr 2003 wegen der zu erwartenden Begutachtung 14.000 DM beantragt. Für Verbrauchsmaterial werden 2.000 DM und für Fachbücher und -zeitschriften jeweils 5.000 DM in Ansatz gebracht.

Außerdem werden wir zwei- bis dreitägige wissenschaftliche Kolloquien veranstalten. Die beantragten Kolloquiumsmittel in Höhe von 20.000 DM sollen für diese Zwecke verwendet werden.

Projektbereich A

**Effiziente Unterstützung von
Softwareentwicklungen und -experimenten**

3 Projektbereich A: Effiziente Unterstützung von Softwareentwicklungen und -experimenten

3.01 Zusammenfassende Darstellung des Projektbereiches

Die zentralen Aufgaben dieses Projektbereichs liegen einerseits in der Erbringung von Dienstleistungen für die übrigen Projektbereiche, andererseits in der (sowohl konzeptionellen als auch systemtechnischen) Integration der in allen Bereichen des SFBs entwickelten, sich ergänzenden Methoden und Werkzeuge. Dienstleistungen werden in Form von konzeptionellen Arbeiten allgemeinen Interesses, Werkzeugen, Erfahrungen aus vergangenen Projekt(phas)en und Beratung zur effizienten Durchführung von Softwareentwicklungen und -experimenten für alle B-, C- und D-Projekte angeboten. Werkzeuge beinhalten Plattformwerkzeuge (z. B. Werkzeuge zur Erstellung von Projektplänen, Prozessmaschinen, Messwerkzeuge und Datenanalysewerkzeuge) und Softwareentwicklungswerkzeuge (z. B. Anforderungsbeschreibungswerkzeuge, Testwerkzeuge). Erstere entstehen überwiegend im A-Bereich des SFB, letztere überwiegend in den übrigen Bereichen. Erfahrungen aus vergangenen Projekt(phas)en beinhalten Modelle (z. B. Aufwandsmodelle sowie Projektpläne), Messdaten und sonstige Erfahrungen (z. B. "lessons learned"). Diese werden in der Erfahrungsdatenbank des SFBs strukturiert verwaltet und zugreifbar gemacht.

Im Rahmen der dritten Förderperiode werden drei verschiedene Arten von Experimenten durchgeführt:

- **Projekt-interne Technologieerprobungen:** Jedes B-, C- und D-Projekt entwickelt neue Beschreibungstechniken und Werkzeuge. Dabei werden die Ergebnisse schrittweise durch experimentellen Einsatz erprobt. Diese Erprobungen liegen alleine in der Verantwortung einzelner Projekte und werden i. allg. ohne Abstimmung mit dem Gesamt-SFB und ohne Unterstützung aus A1 durchgeführt.
- **SFB-öffentliche Technologie-Experimente:** Fertigentwickelte Beschreibungstechniken und Werkzeuge werden auf ihre Verbesserung gegenüber vergleichbaren Alternativen (i. allg. Techniken und Werkzeuge aus dem letzten Baselining) überprüft. Dieser Vergleich muss auf der Basis solider experimenteller Pläne durchgeführt werden, damit die Resultate statistisch haltbar sind. Hierzu erfahren die entsprechenden Teilprojekte Unterstützung aus A1 (Planung, Durchführung von Experimenten). Die Verantwortung für diese Experimente liegt bei den entsprechenden Teilprojekten, die auch für die Ressourcen zuständig sind.
- **SFB-öffentliche Baselining-Entwicklungen:** In regelmäßigen Abständen (zweimal in der nächsten Förderperiode) werden die experimentell erfolgreichen neuen Beschreibungstechniken und Werkzeuge im Rahmen einer kompletten Baseline-Entwicklung erprobt. Dabei ist das Ziel zu validieren, ob diese individuell erfolgversprechenden Techniken und Werkzeuge auch unter 'realen' Projektbedingungen zu positiven Ergebnissen (z. B. Reduktion des

Entwicklungsaufwands, Verbesserung der Qualität, Reduktion der Entwicklungszeit) führen. Die Baseline-Entwicklungen können als 'Scale-Up'-Experimente verstanden werden. Die Verantwortung für diese Baseline-Entwicklungen liegen beim Gesamt-SFB, diese Verantwortung wird innerhalb der Baseline-Entwicklungen stellvertretend durch Teilprojekt A1 wahrgenommen.

Im Teilprojekt A1 erfolgt die Integration aller käuflich erworbenen sowie im Rahmen anderer SFB-Teilprojekte entstehender Werkzeuge und der Betrieb einer Entwicklungs- und Experimentierplattform. Dies umfasst auch die Beratung bei der Planung und Durchführung von Entwicklungen und Experimenten sowie - über die existierenden Handbücher hinausgehende - Anleitungen bei der Benutzung des Labors.

In den Teilprojekten A2 und A3 werden neben dem Erbringen von Infrastrukturdiensten Methoden und Werkzeuge entwickelt, die insbesondere auf eine verbesserte Prozesssteuerung und Produkt-/Erfahrungsdatenverwaltung sowie deren geeignete Integration abzielen.

So entsteht in A2 eine flexible und anpassbare Prozessmaschine zur teilautomatischen Unterstützung von Baseline-Entwicklungsprojekten und Technologie-Experimenten. Dazu soll die in der vergangenen Förderungsperiode entwickelte Prozessmaschine MILOS um Techniken des Wissensmanagements erweitert werden, um somit den strukturierten Zugriff auf Erfahrungswissen auch während der Projektausführung zu ermöglichen. Als initiale Version der Prozessverwaltungskomponente MILOS (A2) wird mit Beginn der Förderperiode eine integrierte Planungs- und Ausführungsunterstützung - einschließlich Messdatenerfassung- bereitgestellt werden. Im Laufe der Förderungsperiode werden zusätzlich Techniken zur Qualitätssicherung auf der Basis von Qualitätsmodellen und sonstigen Erfahrungen ergänzt.

Dies verändert die Benutzerschnittstelle für Softwareentwickler dramatisch: Der Aufruf von Entwicklungswerkzeugen wird durch die Ausführung von Projektplänen ersetzt; der Aufruf von Entwicklungswerkzeugen sowie die Erfassung von Messdaten wird abhängig vom Projektzustand automatisch erfolgen.

In A3 wird das Ziel der Nutzung objekt-relationaler Datenbanktechnologie zur Entwicklung und Bereitstellung eines umfassenden SE-Repositories verfolgt. Dabei soll ein objekt-relationales DBVS (ORDBVS) als Vehikel zur Integration von Prozess-, Produkt- und Erfahrungsdaten genutzt werden. Zunächst wird mit dem Beginn der kommenden Förderperiode ein in Zusammenarbeit von A1 und A3 entwickelter Prototyp einer Erfahrungsdatenbank zur allgemeinen Nutzung bereitgestellt. Dieser wird dann dahingehend weiterentwickelt, dass auch Produktdaten verwaltet werden können, Versionierung und Konfigurierung geeignet unterstützt wird und eine Kopplung mit MILOS hergestellt werden kann. Nur durch diese Integration kann eine umfassende Unterstützung der geplanten Experimente gewährleistet werden. Zur Erreichung dieses Integrationszieles ist eine enge Kooperation mit allen Teilprojekten erforderlich.

Die methodischen Grundlagen von A2 und A3 werden in Zusammenarbeit mit den Projekten B1 und B2 weiterentwickelt. In diesen Teilprojekten entsteht ein generischer Prozessrahmen, der alle Software-Entwicklungsaktivitäten abdeckt. Dabei steht im Vordergrund die Verknüpfung und Integration von existierenden Entwicklungsmethoden und -werkzeugen mit dem Ziel, horizontale und vertikale Verfolgbarkeit von Entwurfsentscheidungen zu gewährleisten. Die Verwaltung/Koordination des laufenden Projekts und somit die Abhängigkeitsverwaltung während der Planung und Ausführung ist ein besonderes Anliegen von B2, während in B1 der Aspekt der Qualitätssicherung durch entsprechende Qualitätsmodelle und -sicherungsprozesse einen besonderen Stellenwert besitzt.

Hierdurch wird auch die Wiederverwendung und Anpassung von Projekt- und Experimentierplänen sowie sonstiger Erfahrungen effizienter gestaltet werden. Zusammen mit den Ergebnissen der Teilprojekte B1 und B2 werden die Ergebnisse von A2 und A3 schrittweise in A1 einfließen und somit dem gesamten SFB zur Verfügung stehen. Damit ist garantiert, dass in dieser nächsten Förderperiode alle Entwicklungen und Experimente optimal unterstützt werden können. Mit dem Grad der Automatisierung von A1 wird der Bedarf an Beratung aus A1 zurückgehen.

3.02 Bisherige und beantragte Förderung des Projektbereiches

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwaltungsausgaben	Investitionen	Gesamt
bis 1997	954,4	29,0	164,4	1147,8
1998	504,0	39,0	-	543,0
1999	516,0	29,0	-	545,0
2000	522,0	29,0	-	551,0
Zwischensumme	2496,4	126,0	164,4	2786,8
2001	891,6	21,0	125,0	1037,6
2002	891,6	21,0	65,0	977,6
2003	891,6	21,0	65,0	977,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000,-)

3.1 Allgemeine Angaben zum Teilprojekt A1

3.1.1 Thema

SE-Labor

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik - Software Engineering

3.1.3 Leiter

Prof. Dr. Dieter Rombach
geb. 6. Juni 1953
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2895
Fax: 0631/205-3331
rombach@informatik.uni-kl.de

Die Stelle des Leiters ist befristet ☒ ja ☐ nein, bis zum

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ ja	☒ nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ ja	☒ nein
Tierversuche	☐ ja	☒ nein
gentechnologische Untersuchungen	☐ ja	☒ nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	954,4	29,0	164,4	1147,8
1998	201,6	23,0	0,0	224,6
1999	206,4	23,0	0,0	229,4
2000	208,8	23,0	0,0	231,8
Zwischensumme	616,8	69,0	0	685,8
2001	351,6	15,0	125,0	491,6
2002	351,6	15,0	65,0	431,6
2003	351,6	15,0	65,0	431,6

(jeweils Ergänzungsausstattung, Beträge in DM 1.000,-)

3.2 Zusammenfassung

Techniken, Methoden und Werkzeuge für die Softwareentwicklung müssen experimentell erprobt werden, um Erfahrungen über ihre Stärken und Schwächen unter verschiedenen Projektbedingungen zu gewinnen. Derartige Erfahrungen sind unerlässliche Voraussetzung für die effektive Wiederverwendung von Techniken, Methoden und Werkzeugen mit wiederholbaren Ergebnissen sowie ihre systematische Verbesserung.

Das Teilprojekt A1 ist ein Infrastrukturprojekt, das die anderen Teilprojekte im Rahmen eines Software-Engineering(SE)-Labors bei der experimentellen Erprobung ihrer Beschreibungstechniken und generischen Methoden unterstützen soll. Dabei übernimmt das Teilprojekt A1 die Rolle des Auftragnehmers zur Unterstützung bei der Durchführung von Experimenten. Unter dem Begriff *Experiment* werden im SFB drei verschiedenen Arten von empirischen Studien verstanden: 1) *projekt-interne Technologieerprobungen*, welche der schrittweisen Erprobung von projekt-internen Ergebnissen dienen, 2) *SFB-öffentliche Technologie-Experimente* (kurz: *Technologie-Experimente*), um fertigentwickelte Beschreibungstechniken und Werkzeuge im Vergleich zu zuvor verwendeten Techniken bewerten zu können, sowie 3) *SFB-öffentliche Baseline-Entwicklungen* (kurz: *Baseline-Entwicklungen*), um im Rahmen einer kompletten Software-Entwicklung die Ergebnisse der Technologie-Experimente überprüfen zu können. Projekt-interne Technologieerprobungen finden in der Verantwortung der jeweiligen Teilprojekte statt und werden daher in diesem Antrag nicht weiter vertieft (s.a. Vorspann Projektbereiche A und B).

Die langfristigen Ziele des Teilprojekts A1 bestehen in der Bereitstellung eines SE-Labors, welches folgende Aspekte systematischer Software-Entwicklungsexperimente unterstützt:

- a) Erstellung expliziter Experimentpläne
- b) Durchführung von Experimenten auf der Basis solcher Pläne
- c) Messdatenerfassung und Datenanalyse während der Experimentdurchführung zum Zwecke der Qualitätssicherung und empirischen Erfahrungsgewinnung
- d) Ablage und Verwaltung von Modellen, Beschreibungstechniken, Produkten und Prozessen gemeinsam mit empirisch gewonnener Erfahrung in einer Datenbank zum Zwecke der Wiederverwendung

In der ersten Förderungsperiode wurde ein initiales SE-Labor auf der Basis industriell käuflicher Entwicklungswerkzeuge aufgebaut. Mit Unterstützung des A1-Personals wurde eine erste Serie von Experimenten geplant und durchgeführt. Diese erste Experimentserie diente im wesentlichen dem Verständnis der Domäne sowie der existierenden Entwicklungsmethoden (z.B. CoDEX). Ergebnisse der Experimente waren u.a. Experimentpläne, Softwaresysteme, Messdaten, Prozess- und Qualitätsmodelle sowie sonstige Erfahrungen, die in entsprechenden experiment-spezifischen Datenbereichen des SE-Kerns zur Verfügung stehen (vgl. ❶ Abb. 1).

In der zweiten Förderperiode wurden mit Hilfe des SE-Labors zwei zentrale Baseline-Entwicklungen sowie darauf aufbauend zahlreiche Technologie-Experimente erfolgreich durchgeführt. Sie dienten einerseits der Bewertung des gesamten Entwicklungsprozesses für die Steuerungssoftware und andererseits der Erprobung erster Techniken und Methoden aus den Teilprojekten. Die aus diesen Experimenten gewonnenen Ergebnisse wurden nach einem einheitlichen Schema im projekt-spezifischen Teil der SFB Erfahrungsdatenbank (SFB-EDB) nachvollziehbar abgelegt. Um die Ergebnisse auch für Folge-Experimente sinnvoll nutzen zu können, wurden diese auf Gemeinsamkeiten und Unterschiede hin untersucht und im entsprechenden Kontext für die Wiederverwendung im experiment-übergreifenden Teil der SFB-EDB abgelegt (vgl. ❷ Abb. 1). Zusätzlich wurden daraus resultierende Anregungen für neue Experimente an alle Teilprojekte geliefert. Die Eignung der Ablagestruktur des experiment-spezifischen Teils der SFB-EDB wurde anhand der Zugriffshäufigkeit auf unterschiedliche SE-Labor-Dienstleistungen sowie aufgrund von Befragungen der Teilprojekte bewertet.

Die Zielsetzung für die dritte Förderperiode stellt die konsequente Weiterführung und Weiterentwicklung des SE-Labors dar. Die Unterstützung der Durchführung von Experimenten wird fortgesetzt. Geplant sind dabei innerhalb des SFB zwei weitere Baseline-Entwicklungen im Bereich der Gebäudetechnik sowie eine Vielzahl spezifischer Technologie-Experimente in den Teilprojekten. Wichtig ist hierbei, dass von diesem Teilprojekt Unterstützung bei der Planung, Durchführung und Analyse von Technologie-Experimenten und Baseline-Entwicklungen geleistet wird, jedoch solche Experimente nicht aktiv initiiert werden. Daher ist es notwendig, die Schnittstellen zu diesem Teilprojekt genau zu definieren: es sollen Szenarien entwickelt werden, die beschreiben, wie bei einer Planung, Durchführung und Analyse eines Experiments mit Unterstützung von Teilprojekt A1 vorzugehen ist. Der weitere Ausbau des SE-Labors bildet weiterhin einen wichtigen Aufgabenbereich, da insbesondere ab der

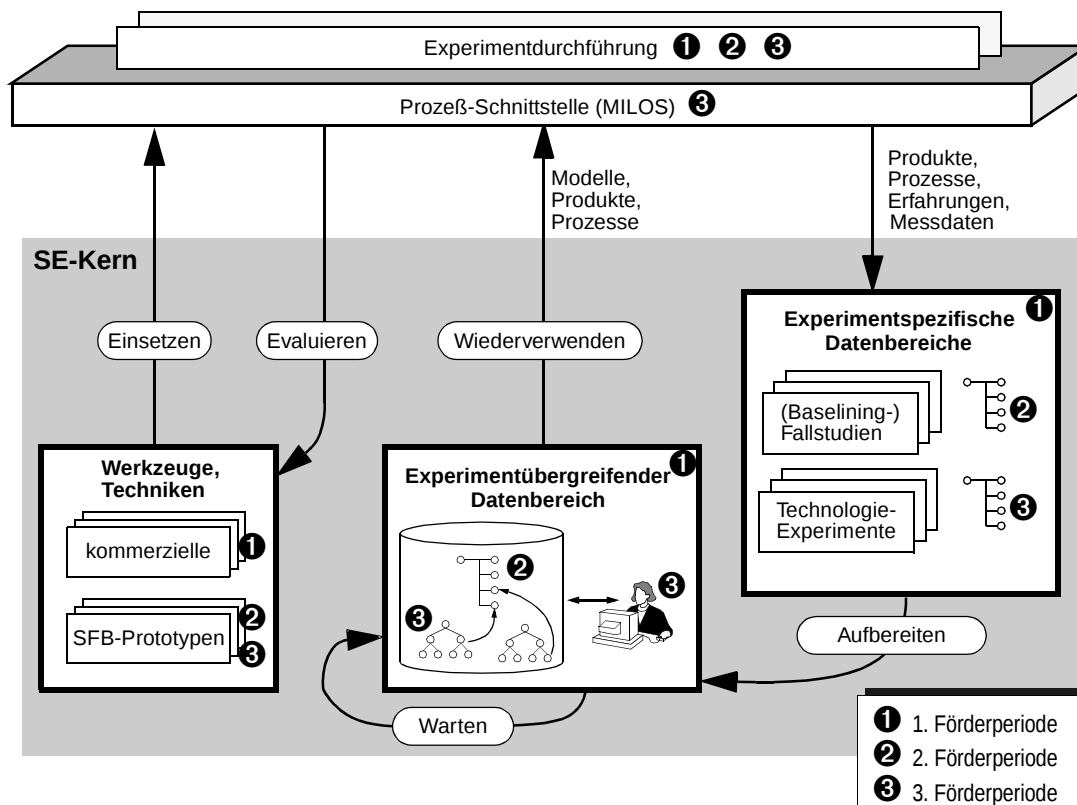


Abbildung 1: SE-Kern

dritten Förderperiode vermehrt die Integration von internen Forschungsprototypen aus den Teilprojekten geplant ist (vgl. ③ Abb. 1).

Die Ablagestruktur der verschiedenen Bereiche der SFB-EDB behandelt derzeit alle wiederverwendbaren Artefakte wie z.B. Experimentpläne oder SDL-Muster für Kommunikationssysteme als 'Black-Boxes', die mit einem Charakterisierungsvektor versehen werden. Für zielgerichtete Analyse und Wiederverwendung der Elemente im Rahmen der Planung und Durchführung neuer Projekte ist das derzeitige Schema jedoch noch zu grob. Die Entwicklung eines feingranulareren Ablage- und Charakterisierungsschemas sowie die Anpassung der Zugriffsgranularität auf Basis einer objekt-relationalen Datenbank stellt daher einen notwendigen Schritt dar, um exakteres Lernen sowie systematische Analyse von SE-Erfahrungen zu ermöglichen. Zudem muss eine kontinuierliche Evaluierung der Inhalte des SE-Labors sowie der SFB-EDB erfolgen.

3.3 Stand der Forschung

Die nachfolgende Veröffentlichungsliste gibt einen kurzen, repräsentativen Überblick über den derzeitigen Stand der Forschung in für das Teilprojekt A1 relevanten Themengebieten. Eine aktuelle Übersicht über derzeit existierende Ansätze und Methoden zum Thema *Experimentieren* gibt [5]. Für die prozess-integrierte Planungs- und Ausführungsplattform PRIME werden in [3] flexible Werkzeugschnittstellen beschrieben. Mit Hilfe von Werkzeug-Modellen sind damit flexible Werkzeug-Akti-

vierungen in Abhängigkeit des Prozessstatus der Ausführungsplattform definierbar. Mit dem Forschungsthema *Erfahrungsdatenbanken/Erfahrungsmanagement-Systeme* beschäftigen sich derzeit mehrere Forschungsgruppen weltweit (siehe [1], [2], [4]). Bei allen Arbeiten liegt der Forschungsschwerpunkt jedoch ausschließlich auf der Verwaltung von wiederverwendbarem Erfahrungswissen. Eine integrierte Verwaltung dieses Wissens zusammen mit den jeweiligen Projektdatenbanken der Experimente (in denen es wiederverwendet bzw. gewonnen wird) ist jedoch, im Gegensatz zum SFB-EDB Ansatz, kein Gegenstand der Forschung.

Literatur

- [1] K.-D. Althoff, A. Birk, S. Hartkopf, W. Müller, M. Nick, D. Surmann, C. Tautz: *Managing Software Engineering Experience for Comprehensive Reuse*; IESE Technical Report 001.99/E, Kaiserslautern, Germany, Fraunhofer Institute for Experimental Software Engineering (IESE), 1999.
- [2] S. Henninger: *An Environment for Reusing Software Processes*; Proc. of the 5th IEEE International Conference on Software Reuse (ICSR5), Victoria, BC, Canada, 1998.
- [3] K. Pohl: *PRIME—Towards Process-Integrated Modelling Environments*; ACM Transactions on Software Engineering and Methodology, Vol8/4, Oktober 1999, pages 343-410.
- [4] R. Webby, C. Seaman, M. Mendonça, V. R. Basili, Y.-M. Kim: *Implementing an Internet-Enabled Software Experience Factory: Work in Progress*; Proc. of the ICSE 99 Workshop on “Software Engineering over the Internet”, online at <http://sern.cpsc.ucalgary.ca/~maurer/ICSE99WS/Program.htm>, May 1999.
- [5] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslen: *Experimentation in Software Engineering. An introduction*; The Kluwer International Series in Software Engineering, Kluwer Academic Publishers, Boston 2000.

3.4 Eigene Vorarbeiten

Die in der kommenden Förderperiode geplanten Arbeiten bauen größtenteils auf den Ergebnissen der vorangegangenen Förderperiode auf. Erste Überlegungen zu der in den Arbeitsschwerpunkten B, C und D geplanten Erweiterungen der SFB-EDB auf der Basis objekt-relationaler Datenbank-Technologie finden sich bereits in [9]. Als Grundlage für die geplante Verfeinerung und Optimierung des SFB-EDB Schemas dienen die existierenden Beschreibungen zur systematischen Analyse und Ablage von Erfahrungswissen (siehe z.B. [6], [8]). [11] beschreibt den Fallstudien-Bereich des experiment-spezifischen Datenbereichs der SFB-EDB, welche die Basis zur Restrukturierung des Bereichs “kontrollierte Experimente” darstellt (s. Arbeitspaket C1). Hinsichtlich einer verbesserten Gestaltung des Zugriffs auf die Inhalte der SFB-EDB

(s. Arbeitspaket D2) existieren bereits erste Überlegungen, die in [7] dargelegt sind. Die Integration von Produktverwaltungsmechanismen in prozessorientierte Umgebungen (s. Arbeitspaket B2) sind Gegenstand einer aktuell laufenden Dissertation [10].

Literatur

- [6] R. L. Feldmann, J. Münch, S. Vorwieger: *Towards Goal-Oriented Organizational Learning: Representing and Maintaining Knowledge in an Experience Base*; Proc. of the 10th International Conference on Software Engineering and Knowledge Engineering (SEKE'98), pages 236–245, San Francisco Bay, CA, USA, June 1998.
- [7] R. L. Feldmann, M. Frey, M. Habetz, M. Mendonça: *Applying Roles in Reuse Repositories*; Proc. of the 12th International Conference on Software Engineering and Knowledge Engineering (SEKE'2000), Chicago, USA, July 2000.
- [8] R. L. Feldmann, B. Geppert, F. Rößler: *An Integrating Approach for Developing Distributed Software Systems -Combining Formal Methods, Software Reuse, and the Experience Base-*; Proc. of the 5th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'99), p.54ff, Las Vegas, Nevada, USA, October 18-22, 1999.
- [9] N. Ritter, H.-P. Steiert, W. Mahnke, R. L. Feldmann: *An Object-Relational SE-Repository with Generated Services*; Proc. of the 1999 Information Resources Management Association International Conference (IRMA99), pages 986–990, Hershey, Pennsylvania, USA, May 1999.
- [10] S. Vorwieger: *Systematische Integration von Prozess- und Produktmanagement*; GI'99, Jahrestagung der Gesellschaft für Informatik 1999, Springer Tagungsband, 5-9. Oktober 1999.
- [11] S. Vorwieger, H. Damczyk, M. Fechtig, B. Schadt, O. Swienty: *Struktur und Werkzeuge des experimentspezifischen Datenbereichs der SFB 501 Erfahrungsdatenbank*; Technical Report SFB 501-11/99, Fachbereich Informatik, Universität Kaiserslautern, 1999.

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Nach dem Aufbau einer initialen Infrastruktur zur Projektunterstützung (1. Förderperiode) und deren Erprobung und Anpassung (2. Förderperiode) stehen nun in verstärktem Maße die Weiterentwicklung der SFB-EDB zur zielgerichteten Analyse und Wiederverwendung von SE-Erfahrungen sowie die intensive Bewertung von SFB-Technologien durch kontrollierte Technologie-Experimente im Vordergrund. Darauf

basierend ergibt sich eine Aufteilung der Aufgaben des Teilprojekts in die folgenden Bereiche:

- Bereitstellung und Erweiterung einer integrierten Plattform für Software Engineering Experimente
- Systematische Analyse und Ablage von Erfahrungen im experiment-übergreifenden Teil der SFB-EDB für kontextspezifische Verwendung während der Planung
- Feingranulare Erfassung von Messdaten
- Unterstützung bei der Durchführung von Software Engineering Experimenten

Diese werden im weiteren genauer erläutert.

Integrierte Plattform für Software Engineering Experimente

Bisher wurden hauptsächlich kommerzielle Werkzeuge zur Unterstützung von Entwicklungsschritten in das SE-Labor integriert. Für die dritte Förderperiode wird eine Vielzahl von SFB-internen Forschungsprototypen aus der jetzigen Förderperiode erwartet. Zentrale Komponenten sind zum einen die in Teilprojekt A2 entwickelte Projektplanungs- und -ausführungsplattform MILOS und zum anderen eine verbesserte Version der SFB-EDB. Die Forschungsprototypen müssen so entwickelt werden, dass sie jederzeit —auch außerhalb von Experimenten— im SE-Labor benutzbar sind. Für das langfristige Ziel, eine vollständig integrierte Plattform für Planung, Ausführung, Analyse und Wiederverwendung bereitstellen zu können, lassen sich drei, sich ergänzende Ziele identifizieren:

a) Unterstützung der Planung und Kontrolle von Experimenten

Die in Teilprojekt A2 entwickelte Prozessmaschine MILOS erweitert das Labor um eine Planungs- und Ausführungskomponente zur Unterstützung von Experimenten. MILOS koordiniert arbeitsteilige Entwicklungsaktivitäten, synchronisiert Entwicklungs- mit Messaktivitäten und kontrolliert Werkzeugaktivierungen. Ebenso unterstützt MILOS Umplanungen und die Revidierung von Entscheidungen während der Abwicklung eines Entwicklungsprozesses, indem kausale Abhängigkeiten verwaltet werden. MILOS stellt jedoch nur Kern-Funktionalitäten bereit. Für eine umfassende Unterstützung des in Teilprojekt B1 entwickelten experimentellen Ansatzes ist die Ergänzung durch weitere Komponenten notwendig. Hierzu zählen die in Teilprojekt B1 entwickelten Werkzeuge zur Generierung von Experiment- und Projektplänen, Werkzeuge zur Erfassung von Messdaten und qualitativer Erfahrung sowie der in Teilprojekt A1 entwickelte Ansatz zur planungs- und prozessorientierten Produktverwaltung.

b) Unterstützung der Analyse und Wiederverwendung

Die SFB-EDB stellt die zentrale Komponente für die Analyse von Experiment-Ergebnissen, deren generische Auslegung und Wiederverwendung dar. Ihre Schnittstellen beschreiben ihr funktionales Spektrum für den jeweiligen

Benutzer und anzubindende Werkzeuge. Für die Ablage und den Zugriff auf die SE-Erfahrungen im experiment-übergreifenden Teil der SFB-EDB wurden im Teilprojekt A1 Ablageschemata und Schnittstellen definiert und durch das Teilprojekt A3 mit Hilfe des objekt-relationalen Datenbankmanagementsystems Informix realisiert. Zur Erweiterung der Funktionalität sollen mächtigere Schnittstellen entwickelt und zur Verfügung gestellt werden.

c) Integration von Technologie-Prototypen

Aus den B- und C-Teilprojekten werden verstärkt Prototyp-Werkzeuge zur Unterstützung generischer Methoden für die Entwicklung großer Systeme in das SE-Labor integriert. Die Integration bezieht sich einerseits auf die Planungs- und Ausführungsplattform, an welche die Prototypen angebunden werden müssen, sowie auf die Fähigkeit, auf SE-Erfahrungen aus der SFB-EDB direkt zugreifen und sie dort wieder einbringen zu können. Hierzu werden explizite Daten- und Prozessschnittstellen in Zusammenarbeit mit den Teilprojekten A2 und A3 zur Verfügung gestellt.

Systematische Analyse und Ablage von Erfahrungen im experiment-übergreifenden Teil der SFB-EDB

In der zweiten Förderperiode wurden verstärkt Experimente (Baseline-Entwicklungen und Technologie-Experimente) im SE-Labor durchgeführt und im experiment-spezifischen Teil der SFB-EDB dokumentiert. Basierend auf diesen Experiment-Dokumentationen müssen die einzelnen Experimente in Beziehung gesetzt und auf Gemeinsamkeiten und allgemein gültige Erkenntnisse hin untersucht werden, die über Experimentgrenzen hinaus gültig sind. Die dabei gewonnenen Erfahrungen werden, soweit sie relevant sind, aufbereitet und anhand von Richtlinien und Dokumentationsstandards beschrieben, und im experiment-übergreifenden Teil der SFB-EDB hinterlegt. Dort stehen sie zukünftigen Experimenten zur Unterstützung bei der Planung und Durchführung zur Verfügung.

Erste Erkenntnisse dieser Art wurden bereits aus den durchgeführten Experimenten abgeleitet und aufbereitet. Mit der steigenden Zahl der Experimente und den daraus resultierenden Informationen im experiment-spezifischen Teil der SFB-EDB wird die derzeitige manuelle und methodisch nur unzureichend unterstützte Ableitung neuer, wiederverwendbarer Erkenntnisse zunehmend erschwert; müssen doch potentiell immer mehr Informationen in die Analyse mit einbezogen werden. Eine systematische und möglichst (teil-)automatisierte Analyse der Experimente ist zu entwickeln und in geeigneter Art und Weise zu unterstützen. Damit einhergehend erfolgt eine Herleitung von projektübergreifendem Wissen, basierend auf der bereits erfolgenden, strukturierten Ablage im experiment-spezifischen Teil der SFB-EDB. Zur besseren Unterstützung der Wiedererwendung der neu gewonnenen Erkenntnisse, müssen diese darüber hinaus in einer einheitlichen und systematischen, ihren Inhalten entsprechend angepassten Repräsentation im experiment-übergreifenden Teil der SFB-EDB persistent eingelagert werden. Dazu ist die derzeit existierende grobe Klassifikation der unterschiedlichen Wiederverwendungsartefakte nach Produkt-Typen um Vor-

lagen mit einheitlichen Charakterisierungsschemata zu erweitern, die einen minimalen Standard garantieren.

Feingranulare Erfassung von Messdaten zur Projektanalyse

In diesem Aufgabenbereich wird die bereits in den vorherigen beiden Förderperioden angesprochene Thematik des Messens und Bewertens als Basis für die systematische Analyse einzelner, zum Zwecke der Erkenntnisgewinnung durchgeführter Experimente erneut aufgegriffen und vertieft. Mit zunehmendem Wissensstand reichen die bisher erhobenen baseline-typischen Messdaten, wie etwa Informationen bezüglich der aufgewendeten Kalenderzeit, des Aufwands für Rework oder eine Typisierung der aufgetretenen Fehler, nicht mehr aus, um weitergehende Hypothesen zu überprüfen und damit verbundene Analysen zu unterstützen. Weitere, experiment-spezifische Messdaten für die im Experiment eingesetzte Entwicklungstechnologie müssen in Zukunft verstärkt erhoben werden. Mit dieser Problematik eng verbunden ist eine geeignete Ablage der vermehrten Messdatenpunkte im experiment-spezifischen Teil der SFB-EDB. Dabei ist jedoch zu beachten, dass die Experimente durch die vermehrte Erfassung von Messdaten nicht unnötig kompliziert und zeitlich ausgedehnt werden. Eine zunehmend automatische Erfassung von charakteristischen Kenngrößen eines Experiments durch geeignete Werkzeuge ist anzustreben. Insbesondere die durch die vermehrt zum Einsatz kommenden prototypischen Werkzeuge aus den einzelnen SFB-Teilprojekten erhobenen Informationen müssen (teil-)automatisch in die SFB-EDB integriert werden, um so für tiefergehende Analysen jederzeit abrufbar vorhanden zu sein. Eine weitgehende Anbindung und/oder Integration der prototypischen Werkzeuge der einzelnen SFB-Teilprojekte an/in die SFB-EDB ist unumgänglich und muss in enger Kooperation mit den einzelnen Teilprojekten vollzogen werden.

Unterstützung von Software Engineering Experimenten

Die systematische Durchführung von Experimenten ist die Basis für zielgerichtetes Lernen und Bewerten von Techniken, Methoden und Werkzeugen. Das Teilprojekt unterstützt dabei in der dritten Förderperiode hauptsächlich Baseline-Entwicklungen im Bereich der Gebäudetechnik und Technologie-Experimente der einzelnen Teilprojekte. Interne Technologieerprobungen liegen im Verantwortungsbereich der jeweiligen Teilprojekte.

In den ersten beiden Förderperioden leistete das SE-Labor hauptsächlich durch die Bereitstellung des technischen und die Experimentiermethodik betreffenden Know-how's Unterstützung bei der Durchführung von SE-Experimenten. In der dritten Förderperiode werden in verstärktem Maße ergänzend verbesserte Techniken und Methoden zur Analyse und Wiederverwendung von SE-Erfahrungen aus vorangegangenen Experimenten eingesetzt. Sie erlauben die Identifikation von Schwachstellen von Methoden und Prozessen in umfassenden Baseline-Entwicklungen. Motiviert durch die Baseline-Ergebnisse können weiterführend kontrollierte Technologie-Experimente zielgerichtet und effizient geplant und durchgeführt werden. Dabei werden einzelne Technologien durch im SFB entwickelte Alternativen ersetzt und zunächst in einer kontrollierten Umgebung deren Verbesserungspotential überprüft. In dem darauf

folgenden Baselining kommen die positiv bewerteten Alternativen zum Einsatz, um die im Technologie-Experiment erzielten Ergebnisse auch in einer allgemeingültigeren Umgebung zu validieren (vgl. Abb. 2).

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Unterstützung und Durchführung von Experimenten

Dieser Arbeitsschwerpunkt stellt die Weiterführung des Arbeitsschwerpunkt C der zweiten Förderungsperiode dar. Bei der Analyse und Auswertung der Baseline-Entwicklungen der zweiten Förderungsperiode stellte sich die Frage, in wieweit die unter universitären Randbedingungen entstandenen Ergebnisse verallgemeinerbar sind. Aus diesem Grund werden ab der dritten Förderungsperiode neben den beiden geplanten Baseline-Entwicklungen im Bereich der Gebäudetechnik und Technologie-Experimenten der einzelnen Teilprojekte auch industrielle Transferprojekte aus anderen Anwendungsbereichen mit externen Kooperationspartnern (z.B. in Zusammenarbeit mit dem Fraunhofer IESE) angestrebt. Für diesen Arbeitsschwerpunkt ergeben sich daher die folgenden Arbeitspakete:

Arbeitspaket A1: Durchführung und Analyse von zwei Baseline-Entwicklungen im Bereich Gebäudetechnik

In diesem Arbeitspaket werden zwei Baseline-Entwicklungen in der Anwendungsdomäne des SFB geplant, durchgeführt und anschließend analysiert. Die Planung erfolgt in der Notation der Modellierungssprache von MILOS, die in der zweiten Förderungsperiode durch Teilprojekt A2 zur Verfügung gestellt wurde, und mit maximalem Wiederverwendungspotential aus der SFB-EDB. Die Durchführung findet jeweils auf der Basis der Pläne in der Prozeßkontrollumgebung MILOS statt (s. Arbeitspaket B1). Die Analyse erfolgt im Hinblick auf die Identifikation von Schwachstellen der eingesetzten Techniken, Methoden und Werkzeuge und wird in die SFB-EDB integriert.

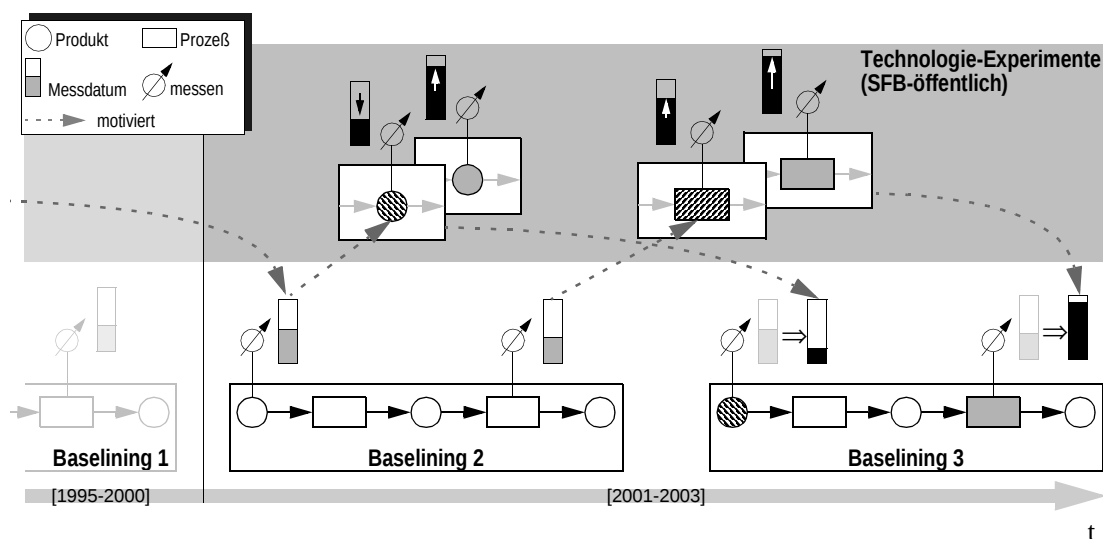


Abbildung 2: Durchführung von Baseline-Entwicklungen und Technologie-Experimenten

Aus den Ergebnissen der Analyse werden nachfolgende Technologie-Experimente (s. Arbeitspaket A2) motiviert.

Baseline-Entwicklungen sind in der Regel mit relativ hohem Aufwand geplante und durchgeführte SE-Experimente. Dies und die Ergänzung durch Technologie-Experimente begründet deren verhältnismäßig niedrige Anzahl im Zeitraum einer Förderperiode.

Ab der dritten Förderperiode sollen SFB-interne Ergebnisse auch in industriellem Umfeld —z.B. durch begleitende Analyse von Transferprojekten— evaluiert werden. Hierdurch sollen praxisrelevante Problemstellungen und Lösungen erarbeitet werden, die in einem rein universitären Umfeld in dieser Form nicht machbar sind. Des Weiteren besteht hier auch die Möglichkeit, in Zusammenarbeit mit Experten einer anderen Anwendungsdomäne Forschungsergebnisse des SFB zu evaluieren.

Personalbedarf: 24 PM

Arbeitspaket A2: Fortlaufende Unterstützung bei der Durchführung und Analyse von Technologie-Experimenten bei allen Teilprojekten zur Evaluierung von Ergebnissen der Teilprojekte

In diesem Arbeitspaket werden Technologie-Experimente für einzelne Teilprojekte im Sinne von kontrollierten Experimenten oder kleineren Fallstudien durchgeführt. Im Gegensatz zur globaleren Fokussierung von Baselines setzen Technologie-Experimente ihren Fokus auf die Untersuchung und Bewertung einer einzelnen, im SFB entwickelten Technologie.

Die Experimente dieses Arbeitspakets werden in enger Zusammenarbeit mit Mitarbeitern der Teilprojekte geplant und durchgeführt, deren Technologien bewertet werden sollen. Dabei steht Personal aus Teilprojekt A1 hauptsächlich zur Unterstützung der Planung und Analyse zur Verfügung. Die Analyse-Ergebnisse werden in die SFB-EDB zur späteren Berücksichtigung für nachfolgende Baseline-Entwicklungen integriert.

Personalbedarf: 33 PM

Arbeitsschwerpunkt B: Ausbau des SE-Labors durch Integration von internen SFB-Teilprojekt-Werkzeugen

Dieser Arbeitsschwerpunkt stellt die konsequente Weiterführung der technischen Dienstleistungen des SE-Labors dar. Während bisher hauptsächlich kommerzielle Werkzeuge zur Unterstützung von Entwicklungsschritten in das SE-Labor integriert wurden, soll in der dritten Förderperiode verstärkt auf SFB-intern entwickelte Prototyp-Werkzeuge eingegangen werden. Eine zentrale Komponente stellt die in Teilprojekt A2 entwickelte Projektplanungs- und -ausführungsplattform MILOS dar, in die alle weiteren SFB-Prototypen integriert werden. Erst durch die Anbindung an MILOS sind Prototyp-Werkzeuge in der kontrollierten Umgebung praktisch einsetzbar. Sie können damit entweder indirekt als Komponente einer Systementwicklung, beispielsweise in einer Fallstudie, oder direkt in einem Technologie-Experiment bewertet werden. Für diesen Arbeitsschwerpunkt ergeben sich die folgenden Arbeitspakete:

Arbeitspaket B1: Integration von MILOS (Teilprojekt A2) zur Unterstützung aller Experimente innerhalb des SFB

In diesem Arbeitspaket wird die im Teilprojekt A2 entwickelte Projektplanungs- und -ausführungsplattform MILOS in das SE-Labor integriert. An diese Plattform sollen während einer Projektdurchführung gemäß des Projektplans die entsprechenden Werkzeuge gebunden werden. Dabei werden sowohl kommerzielle Werkzeuge als auch SFB-Prototypen angebunden. Die resultierende Plattform stellt somit neben der zentralen Projektkontroll-Komponente auch die integrierende Plattform für alle im SE-Labor verfügbaren Werkzeuge dar. Die Integration wird in enger Zusammenarbeit mit Teilprojekt A2 erfolgen.

Personalbedarf: 11 PM

Arbeitspaket B2: Erweiterung von MILOS um Produktmanagement

In der zweiten Förderperiode wurde ein Ansatz entwickelt, der die Aufgaben und Mechanismen des Software-Konfigurationsmanagement in der Planung und Durchführung von SE-Projekten für eine lernende Organisation anpasst, wie sie im SFB realisiert ist. In diesem Arbeitspaket wird der in A1 entwickelte Ansatz in die Prozessplanungs- und -ausführungsumgebung MILOS eingebracht, indem der Produktzugriff auf das Entwicklungs-Repository nicht mehr direkt, sondern über eine Versionierungs-Schicht erfolgt. Desweiteren soll die Prozessplanung um Konzepte des Produktmanagements ergänzt und der Bereich der zu verwaltenden Artefakte auf SE-Erfahrung, d.h. Messdaten zur Projektkontrolle und Analyse, erweitert und in Beziehung zu (Versionen von) Prozessinstanzen gesetzt werden können. Dieser Ansatz ermöglicht einen höheren Grad an Automatisierung und Flexibilität bei der post-mortem-Analyse von Experimenten. Des weiteren sind unter Verwendung der Produktmanagement-Mechanismen Historien über Umplanungen möglich, welche ebenfalls zu Projektanalysen hinzugezogen werden.

Die Tätigkeiten dieses Arbeitspakets sind ab dem zweiten Jahr der beantragten Förderperiode geplant. Es ist zuvor wichtig, MILOS in einem realen Projekt zu evaluieren und eventuelle Verbesserungen in die Plattform einzubringen, bevor eine konkrete Erweiterung dieser Art vorgenommen wird.

Personalbedarf: 15PM

Arbeitspaket B3: Integration der neu entstehenden SFB-Teilprojekt-Werkzeuge

In diesem Arbeitspaket werden Prototyp-Werkzeuge, die in den B- und C-Teilprojekten entwickelt werden, über das SE-Labor allen anderen Teilprojekten sowie allen Projektdurchführungen zur Verfügung gestellt. Beispiele hierfür sind das in Teilprojekt B4 entwickelte Werkzeug zur Definition und Verwaltung von SDL-Mustern oder das B10-Werkzeug zur Verwaltung von Design-Spaces. Hierbei ist einerseits die Anbindung an MILOS vorgesehen, wodurch die Verfügbarkeit während der Durchführung von Projekten gemäß der Planung garantiert werden kann. Andererseits müssen die Prototypen die Tool-Schnittstelle der SFB-EDB bedienen können, um für eine entsprechende Wiederverwendung und Ablage der Werkzeug-Artefakte Sorge tragen

zu können. Die Installation und Integration wird in enger Zusammenarbeit mit den entsprechenden Teilprojekten durchgeführt.

Personalbedarf: 13 PM

Arbeitsschwerpunkt C: Entwicklung eines feingranularen Ablage- und Charakterisierungsschemas für die SFB-EDB auf der Basis objekt-relationaler Datenbank-Technologie

Basierend auf dem durch das Teilprojekt A3 ausgewählten objekt-relationalen Datenbanksystem Informix ist das initiale, mit Hilfe des web-basierten Prototypen der SFB-EDB von A1 entwickelte Ablage- und Charakterisierungsschema für Experimente zu verfeinern. Dazu muss für spezielle, durch einzelne Teilprojekte entwickelte Erfahrungselemente ein darauf zugeschnittener Satz von verfeinernden Attributen bestimmt und in das allgemeine Charakterisierungsschema integriert werden. Des weiteren sind die Anbindung von im SFB 501 entwickelten Werkzeugen an das Erfahrungsdatenbankschema und die darauf zu definierenden Zugriffsschnittstellen zu bestimmen. Es ergeben sich die folgenden Arbeitspakete:

Arbeitspaket C1: Evaluierung und Verfeinerung des Bereichs “kontrollierte Experimente” des experiment-spezifischen Teils der SFB-EDB

In diesem Arbeitspaket erfolgt eine Verfeinerung des bereits jetzt im experiment-spezifischen Teil der Erfahrungsdatenbank enthaltenen Bereichs zur Dokumentation von kontrollierten Experimenten. Dazu wird, ähnlich wie bereits für den Ablagebereich für Fallstudien geschehen, eine Menge von bereits durchgeführten kontrollierten Experimenten systematisch aufbereitet und dokumentiert. Dies geschieht anhand der Schritte der im Teilprojekt B1 entwickelten Experimentier-Methodik. Auf diese Weise wird ein Satz von Kerninhalten bestimmt, der für jedes Experiment zu dokumentieren ist. Des weiteren sind detailliertere Kenngrößen in Form von beschreibenden Attributen in den Charakterisierungsvektor aufzunehmen.

Personalbedarf: 6 PM

Arbeitspaket C2: Wartung und Erweiterung des Charakterisierungsschemas für Erfahrungselemente im experiment-übergreifenden Teil der SFB-EDB

In diesem Arbeitspaket wird das mit Hilfe des web-basierten Prototypen der SFB-EDB entwickelte Charakterisierungsschema für Erfahrungselemente überarbeitet. Dabei sind insbesondere die verbesserten technischen Möglichkeiten zu berücksichtigen, die sich durch den Einsatz des objekt-relationalen Datenbanksystems Informix ergeben. Wurde bisher aufgrund des hohen manuellen Wartungsaufwands nur eine geringe Anzahl von beschreibenden Attributen und Relationen für die einzelnen Bereiche der SFB-EDB verwaltet, so ist es nun möglich, innerhalb dieser Bereiche eine weitere, individuelle Verfeinerung vorzunehmen. Insbesondere können so zum Beispiel auch Beziehungen zwischen einzelnen Erfahrungselementen eines Bereiches berücksichtigt werden. Auf diese Weise wird es möglich, zusammen mit den einzelnen Teilprojekten einen für die jeweiligen Erfahrungselemente zugeschnittenen Cha-

rakterisierungsvektor zu definieren. Zugleich können durch das Datenbanksystem eine Vielzahl von Attributen (teil-)automatisch erfasst, auf Inkonsistenzen geprüft und ggf. global aktualisiert werden.

Personalbedarf: 14 PM

Arbeitspaket C3: Definition und Realisierung verschiedener Sichten zur Erleichterung des Zugriffs auf Erfahrungselemente

In diesem Arbeitspaket werden spezielle auf die Bedürfnisse einzelner Benutzergruppen zugeschnittene Sichten auf die Inhalte der SFB-EDB entwickelt. Dies erfolgt auf der Basis der in den in den Arbeitspaketen C1 und C2 verfeinerten Charakterisierungsvektoren für Erfahrungselemente. Diese Sichten werden dann mit Hilfe des bereits in dieser Förderungsperiode eingeführten und noch zu verfeinernden Rollenkonzepts für die SFB-EDB jedem Benutzer anhand seiner gewählten Rolle zur optimalen Unterstützung bei der Durchführung seiner Aufgaben zur Verfügung gestellt. Durch die Sichtenbildung erfolgt eine für den Benutzer transparente Einschränkung des Erfahrungsdatenbestands auf für ihn relevante und zugängliche Einträge in der SFB-EDB. Sichten dienen somit der Steigerung der Übersichtlichkeit und erlauben zugleich die Definition von Schutzkonzepten, die den Zugriff auf Inhalte der SFB-EDB regeln.

Personalbedarf: 8 PM

Arbeitspaket C4: Anbindung des experiment-spezifischen Teils der SFB-EDB an Werkzeuge zur Produktverwaltung und an das MILOS System

In diesem Arbeitspaket erfolgt eine verfeinerte Modellierung der Daten- und Produktstrukturen des experiment-spezifischen Teils der SFB-EDB. Basierend auf dieser Struktur erfolgt die Kopplung der im Sonderforschungsbereich 501 entwickelten Werkzeuge zur Produktverwaltung und des Projektplanungs- und -ausführungssystems MILOS mit der SFB-EDB und dem darunter liegenden objekt-relationalen Datenbanksystem Informix. Dadurch wird es möglich, die durch die Werkzeuge erfassten und verwalteten Daten und Produkte direkt in den experiment-spezifischen Teil der SFB-EDB zu übernehmen. Sie stehen somit weiteren Auswertungen und Analysen zum Zweck der Erfahrungsgewinnung zur Verfügung.

Personalbedarf: 6 PM

Arbeitspaket C5: Definition von Zugriffsschnittstellen für Werkzeuge im SE-Labor

In diesem Arbeitspaket werden weitere, in den einzelnen Teilprojekten entwickelte Werkzeuge an die SFB-EDB angebunden. Dazu müssen spezielle Zugriffsschnittstellen definiert werden, über die der Datenaustausch zwischen den einzelnen Werkzeugen und den in der objekt-relationalen Datenbank Informix gespeicherten Daten der SFB-EDB ermöglicht. Dieses Arbeitspaket baut auf den Erkenntnissen und Resultaten des Arbeitspakets C4 auf.

Personalbedarf: 12 PM

Arbeitsschwerpunkt D: Systematische Analyse und Ablage von Erfahrungen im experiment-übergreifenden Teil der SFB-EDB für kontextspezifische Verwendung während der Planung

Die zum gegenwärtigen Zeitpunkt definierte Ablagestruktur für wiederverwendbare Artefakte in der SFB-EDB wurde bereits mit Hilfe des Teilprojekts A3 auf das objektrelationale Datenbanksystem Informix übertragen. Durch den Einsatz dieser Technik wird es in Zukunft möglich sein, spezielle Schnittstellen und Eingabemasken zu definieren, die auf die Analyse- und Packaging-Schritte der im Teilprojekt B1 entwickelten Experimentier-Methodik abgestimmt sind. Im Gegensatz zum bisherigen web-basierten Erfahrungsdatenbank-Prototyp können dann Informationen, wie etwa Attribute zur Charakterisierung der Erfahrungselemente, (teil-)automatisch durch das System aus den Experimenten übernommen werden. Durch diese aufgaben-spezifischen Zugriffsschnittstellen erfolgt eine vereinfachte und systematische Ablage von Erfahrungselementen. Zugleich werden so Eingabefehler vermieden, die bisher durch die manuelle Wartung verstärkt auftreten. Zudem können intelligente Analysefunktionen in diese aufgaben-spezifischen Schnittstellen der SFB-EDB eingebaut werden. Daraus ergeben sich die folgenden Arbeitspakete:

Arbeitspaket D1: Vorgabe und Erfassung von für die Analyse zwingend notwendigen Informationen im Charakterisierungsschema der SFB-EDB durch Constraints

In diesem Arbeitspaket wird sichergestellt, dass alle für weiterführende Analysen benötigten Informationen und Messdaten in der SFB-EDB vorhanden sind. Insbesondere sind hierzu Randbedingungen und Regeln zu erstellen, die eine ausreichende Qualität und die Vollständigkeit der erhobenen Daten gewährleisten. Die Berücksichtigung der Regeln bereits in der Planungsphase unterstützt eine rechtzeitige Erhebung aller benötigten Informationen zum Zeitpunkt ihres Entstehens und hilft, nachträgliche Datenerhebungen zu vermeiden. Die Einhaltung dieser Regeln und Vorgaben ist bei der Übertragung der Daten in die SFB-EDB durch das unterliegende objektrelationale Datenbanksystem Informix zu überwachen.

Personalbedarf: 6 PM

Arbeitspaket D2: Definition von intelligenten, aufgaben-spezifischen Schnittstellen für einen verbesserten Zugriff auf die SFB-EDB

In diesem Arbeitspaket werden intelligente Schnittstellen für den Zugriff auf die in der SFB-EDB gespeicherten Daten definiert. Insbesondere sind in diesem Arbeitspaket Schnittstellen zur Unterstützung der Erfahrungsgewinnung durch entsprechende Auswertung und Analyse von Experimentdaten aus dem experiment-spezifischen Teil der Erfahrungsdatenbank zu berücksichtigen. Diese Schnittstellen zielen somit auf die verbesserte Interaktion zwischen Mensch und SFB-EDB durch flexible, ergonomisch angepasste (grafische) Benutzerschnittstellen (GUI).

Personalbedarf: 13 PM

Arbeitspaket D3: (Teil-)automatische Integration von Entwicklungsergebnissen in die SFB-EDB

In diesem Arbeitspaket werden Verfahren zur Integration von Entwicklungsergebnissen in die SFB-EDB untersucht. Ziel ist es dabei, dass Informationen, wie etwa Attribute zur Charakterisierung der Erfahrungselemente, (teil-)automatisch durch das System aus den Experimenten übernommen werden. Durch diese spezifischen Zugriffsschnittstellen erfolgt eine vereinfachte und systematische Ablage von Erfahrungselementen. Zugleich werden Eingabefehler vermieden, die bisher häufig durch die manuelle Wartung bedingt auftraten und so zu Inkonsistenzen und Problemen bei der Auswertung führten.

Personalbedarf: 15 PM

Arbeitspaket D4: (Teil-)automatische Updates der SFB-EDB

In diesem Arbeitspaket wird untersucht, in wieweit eine automatische Auswertungsunterstützung der im SFB durchgeführten und im experiment-spezifischen Teil der SFB-EDB dokumentierten Experimente erfolgen kann. Insbesondere wird dabei an eine Übernahme von z. B. Messdaten und Analyseergebnissen anderer, im SFB entwickelter Werkzeuge gedacht. Dies setzt einen permanenten Informationsaustausch zwischen den durch die Arbeitspakete C4 und C5 an die SFB-EDB gekoppelten Werkzeugen voraus. Die so in den experiment-spezifischen Teil der die SFB-EDB fließenden Daten können anhand vorgegebener Regeln aufbereitet werden, um durch die im Arbeitspaket D1 definierten intelligenten Schnittstellen weiter analysiert zu werden. Das Arbeitspaket zielt damit auf eine verstärkte Interaktion zwischen SFB-EDB und Werkzeugen des SE-Labors ab. Der Beitrag von A1 liegt hierbei primär in der Definition und Koordination der benötigten Strukturen und Abläufe. Eine enge Kooperation mit den für die Werkzeugentwicklung zuständigen Teilprojekten und dem Teilprojekt A3 ist insbesondere im Bereich der technischen Realisierung zwingend erforderlich.

Personalbedarf: 22 PM

Arbeitsschwerpunkt E: Inhaltliche Evaluierung des SE-Labors und der SFB-EDB

Für eine geeignete Unterstützung von Experimenten durch das SE-Labor ist es notwendig, Aussagen über die Akzeptanz der angebotenen Dienste und Werkzeuge zu erhalten. Eine kontinuierliche Erfassung von Nutzungsprofilen bezüglich der installierten Werkzeuge und der SFB-EDB ist daher unumgänglich. Durch die Definition geeigneter Messprogramme soll es ermöglicht werden, derzeit in der SFB-EDB noch fehlendes Erfahrungswissen aufzudecken, um so ggf. die Entwicklung weiterer Techniken und Methoden bzw. weitere Experimente anzustoßen. Für diesen Arbeitsschwerpunkt ergeben sich die folgenden Arbeitspakete:

Arbeitspaket E1: Verfeinerung des Messprogramms zur Kosten/Nutzenanalyse der im SE-Labor installierten Werkzeuge

Durch dieses Arbeitspaket wird das in der gegenwärtigen Förderungsperiode begonnene Arbeitspaket AP-D1 fortgesetzt und erweitert. Bisher wird nur die Anzahl der Werkzeugaufrufe im SE-Labor erfasst. Ziel des Arbeitspakets ist es, darüber hinaus quantitative Aussagen über die Nutzung der einzelnen im SE-Labor bereitgestellten Werkzeuge zu erhalten und darauf basierend eine Kosten- / Nutzenanalyse durchzuführen. Dazu sind die im SE-Labor existierenden Werkzeuge nach Aufgabengebieten zu gruppieren. Für die Werkzeuge einer Gruppe sind dann gezielt weitere Informationen bezüglich der Benutzungsakzeptanz zum Beispiel in Bezug auf die Vollständigkeit des benötigten Funktionalitätsumfangs zu erheben. Darauf basierend können innerhalb dieser Gruppen die einzelnen Werkzeuge hinsichtlich ihrer Nutzung miteinander verglichen werden. Dies ist insbesondere für die im SE-Labor installierten lizenzpflichtigen Werkzeuge von Interesse. Es ist zu erwarten, dass viele dieser Werkzeuge durch die bereits existierenden und in der neuen Förderperiode erwarteten weiteren Eigenentwicklungen von prototypischen, die speziellen Methodiken aus den einzelnen Teilprojekten unterstützenden Werkzeuge, ersetzt werden können.

Personalbedarf: 8 PM

Arbeitspaket E2: Analyse der Anfragen / Zugriffe auf die in der SFB-EDB enthaltenen Artefakte

In diesem Arbeitspaket wird die SFB-EDB in Bezug auf die an sie gestellten Anfragen und die von Ihr gelieferten Ergebnisse untersucht. Insbesondere die von der SFB-EDB derzeit nicht oder nur unzureichend beantworteten Anfragen im Rahmen der Planung und Durchführung von neuen Experimenten geben wichtige Hinweise auf eine fehlende / unzureichende Erfahrungsdatenbasis. Durch die Erkenntnisse aus der Analyse der Erfahrungsdatenbankanfragen ergibt sich die Möglichkeit der gezielten Integration von neuem Wissen in die SFB-EDB. Insbesondere können dadurch weitere Experimente im SFB 501 zur Erkenntnisgewinnung und somit der Schließung der Lücken in der SFB-EDB getriggert werden. Bei Suchanfragen an die SFB-EDB, die zur Lieferung von potentiell wiederverwendbaren Informationen führen, ist zu untersuchen, in wieweit die gelieferten Informationen der Suchanfrage gerecht werden und wie der Benutzer die Qualität dieser Informationen bewertet. Auf diese Weise werden Aussagen über die Qualität der in der SFB-EDB gespeicherten Informationen möglich, und es können ggf. gezielte Maßnahmen zur Qualitätsverbesserung veranlasst werden.

Personalbedarf: 10 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausrüstung Antje von Knethen						
	C1*				D1	D3
		C2			D4	
Ergänzungsausrüstung Raimund L. Feldmann						
	C2		D3		C3	
	B3		D1		D4	
		C4		C5		D2
Stefan Vorwieger	A1	A2		A1	A2	
	B1		B2			
N.N., stud. Hilfskraft						
	A1	A2		A1	A2	
	A1	A2		A1	A2	
	B1		B2			
	B3					
	C2				C3	
		C4		C5		
	D2 + D3 + D4					
	D2 + D3 + D4					
	E1 + E2					

* A1 .. E2 sind Arbeitspakete

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Teilprojekt A1 integriert Prototypwerkzeuge aus allen A- und B-Teilprojekten in das SE-Labor und stellt sie auf diese Weise für Experimente in allen Teilprojekten zur Verfügung. Insbesondere wird in Kooperation mit dem Teilprojekt A3 der Ausbau und die Weiterentwicklung der SFB-EDB vorangetrieben und die Integration des Produktverwaltungssystems MILOS des Teilprojekts A2 vorgenommen. Die experimentelle Vorgehensweise des Teilprojekts B1 wird im SE-Labor umgesetzt. Die im experiment-übergreifenden Teil der SFB-EDB abgeleiteten Informationen werden allen Teilprojekten für weitere Experimente bereitgestellt. Mit diesen Teilprojekten ist deshalb ein intensiver Erfahrungsaustausch über die benutzten Beschreibungs- und Generierungstechniken vorgesehen.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	2	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
			StdHkr.	7	142.800	StdHkr.	7	142.800	StdHkr.	7	142.800
	zus.:	2	zus.:	9	351.600	zus.:	9	351.600	zus.:	9	351.600
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		15.000	(522)		15.000	(522)		15.000
			zusammen		15.000	zusammen		15.000	zusammen		15.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			125.000			65.000			65.000		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Rombach, Prof. Dr. (2) Antje von Knethen (3) Althoff, Dr.	Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL Fraunhofer IESE	5 20 B	1.1.95	
Ergänzungsausstattung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Feldmann, Dipl.-Inform. (2) Vorwieger, Dipl.-Inform. (3) N.N., StdHkr. (4) N.N., StdHkr. (5) N.N., StdHkr. (6) N.N., StdHkr. (7) N.N., StdHkr. (8) N.N., StdHkr. (9) N.N., StdHkr.	Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	38,5 38,5 19 19 19 19 19 19 19	1.3.97 1.1.95	IIa IIa
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Der Mitarbeiter der Grundausrüstung verfeinert und evaluiert hauptsächlich den Bereich der “kontrollierten Experimente” im experiment-spezifischen Teil der SFB-EDB und ist für die Bearbeitung der Arbeitspakete C1, C2, D1, D3 und D4 vorgesehen.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Der erste wissenschaftliche Mitarbeiter erarbeitet die konzeptionellen Grundlagen für die verbesserte Version der SFB-EDB. Zugleich ist er mit der Anbindung und Integration von Werkzeugen (u. a. MILOS) an die SFB-EDB beschäftigt. Er ist für die Bearbeitung der Arbeitspakete B3, C2, C3, C4, C5, D1, D2, D3 und D4 vorgesehen.

Der zweite wissenschaftliche Mitarbeiter führt die zentralen Arbeiten zur Unterstützung und Durchführung von Experimenten (Baseline-Entwicklungen und Technologie-Experimente) aus. Zudem wird die Integration des Prozess- und Produktmanagements durch diesen Mitarbeiter vorangetrieben. Folglich ist er für die Bearbeitung der Arbeitspakete A1, A2, B1 und B2 vorgesehen.

Die beantragten wissenschaftlichen Hilfskraftstellen dienen der Unterstützung bei den umfangreichen Implementierungsarbeiten in Bezug auf die Wartung der SFB-EDB und die vorgesehenen Evaluierungstätigkeiten. Die ersten beiden studentischen Hilfskräfte sind dabei zur Unterstützung bei der Durchführung von Experimenten im Arbeitsschwerpunkt A vorgesehen. Die dritte studentische Hilfskraft unterstützt die Implementierungsaufgaben zur Integration von Werkzeugen im SE-Labor (Arbeitspakete B1, B2, B3). Die vierte studentische Hilfskraft führt Tätigkeiten im Rahmen der Arbeitspakete C2-C5 durch. Die fünfte und sechste studentische Hilfskraft ist für Aufgaben im Bereich der Arbeitspakete D2, D3 und D4 vorgesehen. Die siebte studentische Hilfskraft kümmert sich um die inhaltliche Evaluierung des SE-Labors und der SFB-EDB (Arbeitspakete E1 und E2).

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausstattung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausstattung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausstattung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	15.000	15.000	15.000

Begründung zur Ergänzungsausstattung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausstattung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial für das Teilprojekt sowie zur kurzfristigen Beschaffung von kleineren Ersatzteilen für das SE-Labor benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
Fortführende Lizenzen und Beschaffung neuer Werkzeuge	65.000	65.000	65.000
Ersatz von Hardware	60.000		
Summe (entspricht den Summen "Investitionen" in Übersicht 3.7)	125.000	65.000	65.000

Begründung der Investitionen

Die beantragten Beträge für Investitionen sind zum einen zur Fortführung bestehender und zur Anschaffung neuer Lizenzen notwendig. Diese dienen der Versorgung aller Teilprojekte des SFB mit kommerziellen CASE-Werkzeugen und Editoren. Zum anderen müssen im ersten Jahr der neuen Förderperiode die schon jetzt bereits veralteten vierzehn X-Terminal-Clients (Base und Monitor) im zentralen SE-Labor ersetzt werden. Dies ist notwendig, da das SE-Labor in der neuen Förderperiode —wie auch in den vergangenen Förderperioden— intensiv für Baselineing- und Technologie-Experimente in Anspruch genommen werden wird.

3.1 Allgemeine Angaben zum Teilprojekt A2

3.1.1 Thema

Entwicklung einer flexiblen Modellierungs- und Ausführungsumgebung
für Software-Entwicklungsprozesse

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Software Engineering, AG Künstliche Intelligenz

3.1.3 Leiter

Prof. Dr. Michael Richter
geb. 21.6.1938
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2800
Fax: 0631/205-3357
richter@informatik.uni-kl.de

Prof. Dr. Dieter Rombach
geb. 6.6.1953
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2895
Fax: 0631/205-3331
rombach@informatik.uni-kl.de

Die Stellen der Leiter sind befristet ☒ nein ☐ ja, bis zum

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ ja	☒ nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ ja	☒ nein
Tierversuche	☐ ja	☒ nein
gentechnologische Untersuchungen	☐ ja	☒ nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
1998	100,8	3,0	0,0	103,8
1999	103,2	3,0	0,0	106,2
2000	104,4	3,0	0,0	107,4
Zwischensumme	308,4	9,0	0,0	317,4
2001	270,0	3,0	0,0	273,0
2002	270,0	3,0	0,0	273,0
2003	270,0	3,0	0,0	273,0

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Ziel dieses Teilprojekts ist die Entwicklung eines Werkzeugs zur integrierten Unterstützung von management- und technisch-orientierten Prozessen im Rahmen von Software-Entwicklungsprojekten des SE-Labors sowie dessen experimentelle Erprobung. In der nächsten Förderungsperiode soll dazu die systematische Wiederverwendung von Erfahrungen (Prozessmodelle, Qualitätsmodelle etc.) unter Einbezug von domänenspezifischem Hintergrundwissen unterstützt werden. Ein neuer, im SFB noch nicht behandelter, Aspekt ist dabei die Wiederverwendung während der Projektausführung. Hierzu sollen grundlegende Techniken des Wissensmanagements herangezogen werden, in die in der vergangenen Förderungsperiode in Teilprojekt A2 entwickelten MILOS¹ Ausführungs- und Modellierungsumgebung integriert und somit auch anderen Teilprojekten als Dienst zur Verfügung gestellt werden.

Erfahrungen stehen in Form bewerteter Artefakte in der SFB-EB zur Verfügung. Um einen möglichst hohen Nutzen der Wiederverwendung während der Projektausführung zu erreichen, ist es wichtig, auf Erfahrungen situationsspezifisch, d. h. den Zustand des ausgeführten Projekts berücksichtigend, zuzugreifen und sie geeignet in den Entwicklungsprozess einzubinden. Hierzu wird dynamisches Kontextwissen (aktueller Projektzustand, Entscheidungsketten etc.) benötigt, das im Rahmen dieses Teilprojekts und in Zusammenarbeit mit anderen Teilprojekten identifiziert, formalisiert und verwaltet werden soll. Zusätzlich ist auch der Zeitpunkt festzustellen, an dem eine Entscheidung zu treffen ist, in die Erfahrungswissen einbezogen werden kann. Hierdurch ist es möglich, eine Unterstützung pro-aktiv, d. h. selbständig die Initiative ergreifend, zu konzipieren, die der Nichtberücksichtigung von vorhandenen Erfahrungen entgegenwirkt. Um dies zu erreichen, muss das zu realisierende Wissensmanagement in der Lage sein, auf Ereignisse zu reagieren, wie sie z. B. von dem in Teilprojekt B1 zu entwickelnden Prozessleitstand oder der in MILOS integrierten Abhängigkeitsverwaltung generiert werden. Weiterhin müssen Informations-

1. Modeling Language and Operational Support for Software Processes

bedürfnisse ermittelt, modelliert und diagnostiziert werden, die einen Hinweis auf benötigtes Wissen geben, auf dessen Basis Maßnahmen als Reaktion auf entsprechende Ereignisse zu treffen sind. Die Etablierung des hierzu benötigten prozessorientierten Wissensmanagements stellt ein zentrales Ziel dieses Antrags dar. Da das Wissensmanagement möglichst viel dynamische und statische Kontextinformationen berücksichtigen muss, soll eine Anbindung an den Prozessleitstand erfolgen, dessen bewertete Übersicht über den aktuellen Projektzustand ermöglicht, den Eintritt wichtiger Ereignisse (wie z. B. Nichteinhalten von Qualitätsvorgaben) zu ermitteln. Weitere Vorhaben sind die Erfassung und Verwaltung von domänenspezifischem Hintergrundwissen über den strukturierten und situationsspezifischen Zugriff auf andere heterogene Wissensquellen.

Die Ergebnisse dieses Teilprojekts sollen in die MILOS Ausführungs- und Modellierungsumgebung integriert und die Nützlichkeit anhand von Experimenten gezeigt werden. Es wird erwartet, dass durch die hier angestrebte Integration von Wissensmanagement der Anteil von Erfahrungswissen als Entscheidungsgrundlage für das Projektmanagement signifikant höher ausfällt als ohne diese Unterstützung.

3.3 Stand der Forschung

Das Projekt A2 war ursprünglich nur auf eine Förderungsperiode ausgelegt. Die beantragte Fortsetzung, die sich mit der Integration von Techniken des Prozess- und Wissensmanagements befasst, resultiert aus neuen Perspektiven, die sich in der vorangegangenen Förderungsperiode ergaben. Lediglich der diesbezügliche Stand der Forschung soll an dieser Stelle dargelegt werden.

Wissensmanagement umfasst in seiner allgemeinsten Form alle notwendigen Maßnahmen zur Etablierung einer Kultur im Umgang mit Wissen in einer Unternehmung. In betriebswirtschaftlichen Publikationen (z. B. [1], [2]) wird darunter zunächst ein Führungsstil verstanden, der zwar auch soziale Aspekte einschließt, sich aber vor allem mit der Aufdeckung und Nutzbarmachung von Erfahrungswissen befasst, das ansonsten als nicht ausreichend genutztes Potential in den Köpfen der Mitarbeiter existiert. Im Bereich der künstlichen Intelligenz beschäftigt man sich im Rahmen der Erforschung solcher *Organizational Memories* [3] vor allem mit Problematiken des Knowledge Engineerings, also Fragen der Erfassung, Ablage, Wartung und Retrieval von Wissen. Im *Experience Factory* Ansatz [4] wird eine eigene organisatorische Einheit zur Verwaltung von Erfahrungswissen vorgeschlagen, das in einer Erfahrungsdatenbank (EDB) zentral gesammelt werden soll.

In den bisher publizierten Ansätzen zur Nutzung einer Erfahrungsdatenbank (z. B. [12], [17]) wird diese jedoch als passives Repository gesehen, das über ein zentrales Portal dem Benutzer die Navigation zu dem für ihn relevanten Teil sowie die Spezifikation der Suchanfragen überlässt. Die dabei eingesetzten Suchverfahren reichen von manueller Navigation und Entscheidungsbäumen [14] bis hin zur Suche auf strukturierten Wissenseinheiten. Zu letzteren sind SQL-basierte [17], ähnlichkeitsbasierte

[12] sowie ontologie-basierte Ansätze [6] zu zählen. Insbesondere der ontologie-basierte Ansatz passt gut zu modernen Methodologien zur Konstruktion von Wissensbasen (z. B. [5]), die diesen Prozess mehr als Modellierungsprozess statt als Transferprozess verstehen. Ontologien dienen dabei als konzeptionelles Modell zur Beschreibung von Wissen und sind somit für die Suche bereits vorhanden. Die Wissensseinheiten selbst können dabei in strukturierten Dokumenten oder Datenbanken abgelegt sein. In [8] sind dies XML Dokumente. Die Ontologie ist in F-Logic, einer an die Framelogik angelehnten Beschreibungssprache, niedergeschrieben.

Empirische Studien (z. B. [16]) in Unternehmen haben gezeigt, dass einer der maßgeblichen Hinderungsgründe für die Wiederverwendung von in Wissensbasen abgelegten Erfahrungen das Problem ist, in der jeweiligen Situation zu wissen, welche jeweils wiederverwendbaren Erfahrungen vorhanden sind. Die hiermit geforderte proaktive Unterstützung lässt sich durch die Integration von Techniken des Wissens- und Prozessmanagements erreichen, wie sie z. B. in [6] und [7] vorgeschlagen wird. Insbesondere stellen die Beschreibungen von Prozessen (Prozessmodelle) selbst ontologisches Wissen dar, das als Kontext für andere Aktivitäten dient, in denen mit Wissen umgegangen wird.

In [15] wird eine aufgabenspezifische, pro-aktive Suche für den Spezialfall des fallbasierten Retrievals im Rahmen eines Designprozesses vorgestellt. Die dort beschriebene Anwendung integriert eine fallbasierte Suche mit einer Schlüsselwortsuche in Online-Dokumenten und erlaubt nachfolgenden Aufgaben den automatischen Zugriff auf Informationen über vorangegangene Aufgaben, insbesondere auf die dort in Betracht gezogenen früheren Fälle. Die einzelnen auszuführenden Aufgaben sowie die jeweils in ihrem Kontext durchzuführende Informationssuche lassen sich jedoch nicht explizit modellieren. Ein erforderliches allgemeines ähnlichkeitsbasiertes Dokumenten-Retrieval fehlt bisher noch weitgehend.

In [11] ist beschrieben, wie die Lösung des *Capability Matching* Problem bei Workflow-Management-Systemen durch die Integration verschiedener Wissensquellen, welche die Organisationsstruktur, die Fähigkeiten und die Autoritäten von Agenten beschreiben, unterstützt werden kann. Die Organisationsstruktur der Unternehmung ist dabei in einer standardisierten Ontologie [9] modelliert, Autoritäten hingegen mit Hilfe der in [10] vorgeschlagenen Logik beschrieben, deren Nutzen bereits in vielen CSCW- (Computer Supported Cooperative Work-) Systemen gezeigt wurde. Auch hier ist die Unterstützung nicht flexibel sondern speziell für o. g. Problemstellung konzipiert.

Kommerzielle Workflow-Management-Systeme (z. B. TeamWare Flow [<http://teamware.fujitsu.com.au/teamware/Products/>]) unterstützen den Aufruf externer Applikationen mittels Skriptsprachen (z. B. TCL). Hierdurch ließe sich prinzipiell auch eine Informationssuche starten; eine Strukturierung mehrerer solcher Skripte anhand unterschiedlicher Informationsbedürfnisse ist hierdurch jedoch nicht möglich. Zudem könnte durch einen solchen Ansatz nur die Informationssuche während der Ausführ-

runge eines Projektplanes, nicht aber während des Planungsprozesses unterstützt werden.

- [1] I. Nonaka, H. Takeuchi: *The Knowledge Creating Company*, Oxford University Press, 1995
- [2] P. Senge: *The Fifth Discipline Fieldbook: Strategies and Tools for Building a Learning Organization*, Doubleday, 1995
- [3] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, M. Sintek: *Toward a Technology for Organizational Memories*, IEEE Intelligent Systems, 13(3), May/June 1998
- [4] V. R. Basili, G. Caldiera, H. D. Rombach: *Experience Factory*, Encyclopedia of Software Engineering (J. J. Marciniak, ed.), vol. 1, John Wiley Sons, 1994
- [5] R. Studer, D. Fensel, S. Decker, V.R. Benjamins: *Knowledge Engineering: Survey and Future Directions*, F. Puppe (ed.): Knowledge-based Systems: Survey and Future Directions, Proceeding of the 5th German Conf. on Knowledge-based Systems, Wuerzburg, March 1999, Lecture Notes in Artificial Intelligence (LNAI), vol. 1570, Springer-Verlag, 1999
- [6] H. Schnurr, S. Staab, R. Studer: *Ontology-based Process Support*, Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning (AAAI-99). Technical Report, Menlo Park: AAAI.
- [7] S. Staab, H. Schnurr: *Knowledge and Business Processes: Approaching an Integration*, OM '99 - Proceedings of the international Workshop on Knowledge Management and Organizational Memory (IJCAI-99). Stockholm, Sweden, 1999
- [8] M. Erdmann, R. Studer: *Ontologies as Conceptual Models for XML Documents*, Proceedings of the 12th Workshop for Knowledge Acquisition, Modeling and management (KAW'99), Banff, Canada, October 1999
- [9] M. Uschold, M. King, S. Moralee, Y. Zorgios: *The Enterprise Ontology*, The Knowledge Engineering Review, Vol. 13, 1998
- [10] R. Wieringa, J. Meyer: *Applications of Deontic Logic in Computer Science: A Concise Overview*, Deontic Logic in Computer Science, eds R. Wieringa, J. Meyer, Wiley, 1993
- [11] P. Jarvis, J. Stader, A. Macintosh, J. Moore, P. Chung: *What Right Do You Have to Do That? Infusing Adaptive Workflow Technology with Knowledge about the Organisational and Authority Context of a Task*, Proceedings of First International Conference on Enterprise Information Systems (ICEIS-99), Setubal, Portugal, March 1999

- [12] K.-D. Althoff, A.. Birk, S. Hartkopf, W. Mueller, M. Nick, D. Surmann, C. Tautz: *Managing Software Engineering Experience for Comprehensive Reuse*, 11th Int. Conf. on Software Engineering & Knowledge Engineering, 1999, ISBN 1-891706-01-2
- [13] S. Henninger: *Tools Supporting the Creation and Evolution of Software Development Knowledge*, Twelvth International Conference on Automated Software Engineering, Lake Tahoe, NV, Nov., 1997, pp. 48-53.
- [14] S. Henninger: *An Environment for Reusing Software Processes*, Fifth International Conference on Software Reuse, Victoria, British Columbia, Canada, 1998, pp. 103 - 112.
- [15] D. Leake, L. Birnbaum, K. Hammond, C. Marlow, H. Yang: *Task-Based Knowledge Management*, Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning (AAAI-99). Technical Report, Menlo Park: AAAI.
- [16] S. Mahe, C. Rieu: *Toward a Pull-Approach of KM for Improving Enterprise Flexibility Responsiveness: A Necessary First Step for Introducing Knowledge Management in Small and Medium Enterprises*. In: Proceeding of the International Symposium on Management of Industrial and Corporate Knowledge (ISMICK '97), Compiègne, 1997.
- [17] R. Webby, C. Seaman, M. Mendonca, V. Basili, Y. Kim: *Implementing an Internet-Enabled Software Experience Factory: Work in Progress*, Web proc. of the 2nd Workshop on "SE over the Internet" held at ICSE-99, <http://sern.ucalgary.ca/~maurer/ICSE99WS/Submissions/Webby/Webby.html>.

3.4 Eigene Vorarbeiten

Hier wird auf den Ergebnisbericht des Teilprojekts A2 bzw. auf [22] verwiesen. Im folgenden soll lediglich auf für die neuen Aspekte relevante Vorarbeiten eingegangen werden.

Das Wiederfinden von Wissen unter Berücksichtigung der Relevanz für eine gegebene Aufgabe ist eine grundlegende Problematik beim prozessorientierten Wissensmanagement. Hierfür stellen Techniken des fallbasierten Schließens (CBR) einen möglichen Ansatz dar und wurden in der Arbeitsgruppe von Professor Richter ausgiebig erforscht ([19], [20]). In dem Projekt INRECA II, einem EU Projekt (Partner war hier u. a. die Universität Kaiserslautern, AG Richter), wurden die Methoden des Software Engineerings, wie sie auch Gegenstand des SFB 501 sind, auf die industrielle Entwicklung von CBR Systemen (im Wesentlichen aus dem Helpdesk Bereich) angewandt. Die entstandene Methodologie ist in [18] publiziert.

Die Konzeption von Assistenzsystemen zum Zweck der wissensbasierten Unterstützung wurde bereits im Rahmen des Projekts *Intelligenter Bebauungsplan*, einem interdisziplinären Kooperationsprojekt, dem auch die Arbeitsgruppe von Professor Richter angehörte, behandelt (siehe [21]).

Ansätze, die sich speziell mit der Integration von Prozess- und Knowledge Management für Software-Entwicklungsprozesse befassen, wurden in [23] und [24] beschrieben. Diese Publikationen zeigen das Potential des Konzepts der taskzentrierten Modellierung von Prozessen, das auch MILOS zugrundeliegt, zur Unterstützung wissensintensiver Tätigkeiten.

- [18] R. Bergmann, S. Breen, M. Göker, M. Manago, S. Wess: *Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology*, Lecture Notes in Artificial Intelligence, State-of-the-Art-Survey, LNAI 1612, Springer Verlag, 1999
- [19] M. M. Richter, K.-D. Althoff : *Similarity and Utility in Non-Numerical Domains*, Mathematische Methoden der Wirtschaftswissenschaften, Physika-Verlag, pp. 403 – 413, 1999
- [20] K.-D. Althoff, W. Wilke: Potential Uses of Case-Based Reasoning in Experience Based Construction of Software Systems and Business Process Support, R. Bergmann & W. Wilke (eds.), Proc. of the Fifth German Workshop on Case-Based Reasoning, 31-38, 1997
- [21] G. Pews: *Flexible Unterstützung für Entwurfsprozesse im Kontext von Verwaltungsaufgaben*, Dissertation, Fachbereich Informatik, Universität Kaiserslautern, Juli 1999
- [22] F. Maurer, B. Dellen, F. Bendeck, S. Goldmann, H. Holz, B. Kötting, M. Schaaf: Merging project planning and web-enabled dynamic workflow for software development, *IEEE Internet Computing, Special Issue on Internet-Based Workflow, May/June 2000 (to appear)*
- [23] F. Maurer, H. Holz: *Process-Centered Knowledge Organization for Software Engineering*, Aha, D.W., & Munoz Avila, H. (Eds.): Exploring Synergies of Knowledge Management and Case-Based Reasoning: A 1999 AAAI Workshop (Working Notes). (Technical Report AIC-99-008). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, 1999
- [24] F. Maurer, H. Holz: *Process-Oriented Knowledge Management For Learning Software Organizations*, Proceedings of the 12th Knowledge Acquisition Workshop (KAW '99), Banff, Canada, 1999

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Generell beinhalten Software-Entwicklungsprozesse in hohem Grad kreative Aktivitäten, sind nicht-deterministisch, nebenläufig, verteilt und unterliegen oft Änderungen, die eine detaillierte Vorplanung erschweren. Während der Projektdurchführung, d. h. nach Ende der Vorplanung, erfordern Planabweichungen (z. B. wegen Krankheit oder nicht erreichter Qualität) Reaktionen durch das Projektmanagement, wobei eventuell zuvor gemachte Erfahrungen zur Entscheidungsfindung herangezogen werden sollten. Dadurch ergeben sich neue Anforderungen an eine umfassende Unterstützung der Wiederverwendung für ein Software-Entwicklungsprojekt. Die bisher im SFB betriebene Wiederverwendung von Erfahrungen fand überwiegend offline während der Vorplanung statt (siehe Abbildung 1). In der nächsten Förderungsperiode soll Wiederverwendung darüberhinaus online, d. h. während der Abwicklung von Projekten unterstützt werden.

Die zielgerichtete Reaktion bei Abweichungen vom Projektplan und dessen situationsspezifische Verfeinerung verlangen zu ihrer sachgerechten Ausführung erhebliches Wissen (Erfahrungen, domänenspezifisches Hintergrundwissen etc.). Dieses Wissen muss zunächst einmal identifiziert und, falls nicht schon grundsätzlich vorhanden, erworben werden. Es soll dann den richtigen Personen bzw. Stellen zum richtigen Zeitpunkt zur Verfügung gestellt werden. Wesentlich ist hierbei, dass dies systematisch geschieht und insbesondere einmal erhobenes Wissen auch geeignet wiederverwendet wird.

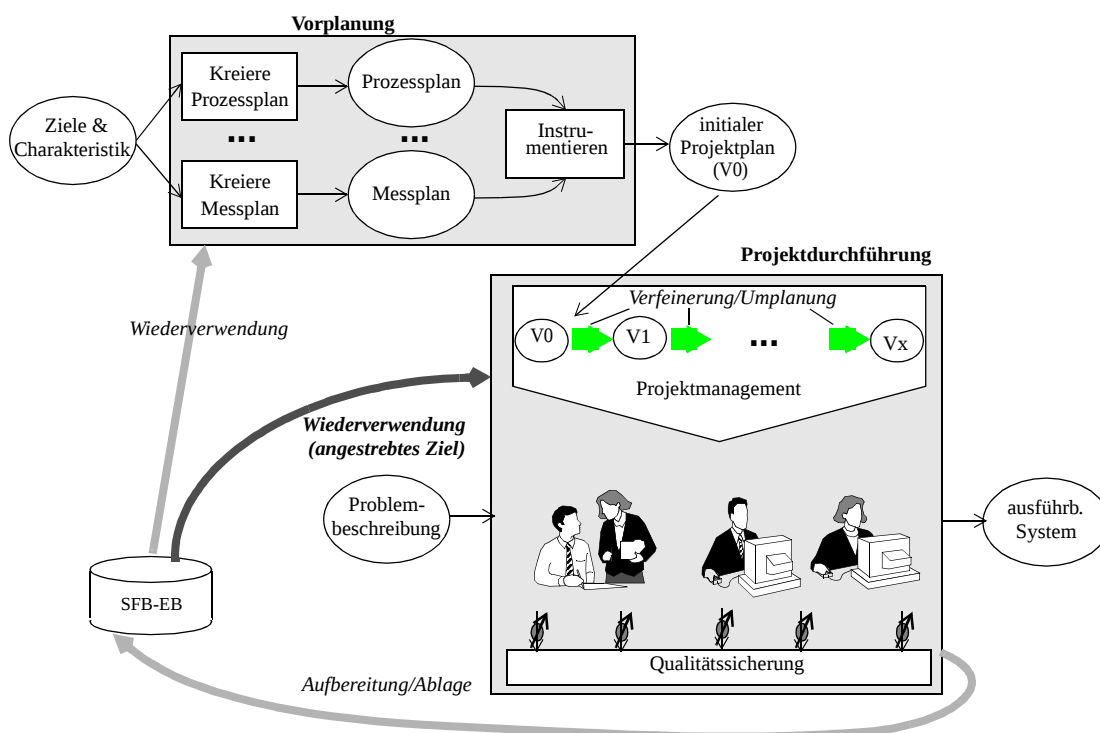


Abbildung 1: Wiederverwendung bei Vorplanung und Umplanung während Projektausführung

Ziel des Teilprojekts A2 in dieser Förderungsperiode ist somit aufbauend auf den Ergebnissen der ersten Förderungsperiode die wissensbasierte Assistenz zur Entscheidungsfindung. Zur Erreichung dieses Ziels sollen Techniken des Wissensmanagements zur umfassenden Unterstützung der Wiederverwendung von Erfahrungen sowie Verwaltung von domänenspezifischem und situativem Wissen in MILOS integriert werden. Erfahrungen sind hierbei Pläne, Modelle und Traces vergangener Projekte, die in bewerteter (geeignet annotierter) Form durch die SFB-EB zugreifbar sind. Es sollen aber auch andere Wissensquellen, wie bereits vorhandene Datenbanken und Dokumenten-Management-Systeme etc. einbezogen werden, die zur Entscheidungsunterstützung auf Basis des aktuellen Projektzustands relevante Informationen beinhalten. Ein besonderer Schwerpunkt des Wissensmanagements soll auf dessen Fähigkeit liegen, nicht nur auf Benutzerintervention zu reagieren, sondern pro-aktiv Vorschläge zur Entscheidungsfindung anzubieten.

Um dies zu ermöglichen, muss auf Ereignisse, die potentiell ein Eingreifen des Projektmanagements erfordern, auch von Seiten des Wissensmanagementsystems reagiert werden. Die Benachrichtigung über den Eintritt eines solchen Ereignisses soll zunächst durch Ankopplung der im Rahmen des Teilprojekts B2 in der vergangenen Förderungsperiode entwickelten Änderungsmanagement-Komponente erfolgen. Im weiteren Projektverlauf ist dann geplant, den in Teilprojekt B1 zu entwickelnden Prozessleitstand hinzuzuziehen, der eine ständige Bewertung des Projektzustands vornimmt und somit über ein breites Spektrum an Ereignissen (z. B. Nichteinhalten von Prozessqualität) Aufschluss gibt. Als Reaktion auf diese Ereignisse soll das zu realisierende Wissensmanagement auf der Basis zuvor modellierter und klassifizierter Informationsbedürfnisse relevante Fragen aufwerfen und daraus Vorschläge zur Entscheidungsfindung aus den vorhandenen Erfahrungen ermitteln. Ergebnisse sollen verantwortlichen Benutzern (z. B. Projektmanagern), welche letztendlich zielgerichtet auf entsprechende Ereignisse reagieren müssen, in pro-aktiver Weise zur Verfügung gestellt werden. Der hierbei erwartete Nutzen ist:

- Das Projektmanagement bekommt einen schnelleren Überblick über entscheidungsrelevante Informationen zur zielgerichteten Reaktion auf Ereignisse.
- Vorhandene Erfahrungen haben bessere Chancen, im Entscheidungsprozess berücksichtigt zu werden.
- Zuvor gemachte Bewertungen von Wissen können den aktuellen Umständen angepasst werden.

Das resultierende MILOS System soll dem SFB als Service zur Verfügung gestellt werden, um dessen Nützlichkeit im Rahmen von Experimenten im SE-Labor zu demonstrieren. Hiermit einhergehend soll die Weiterentwicklung und Anpassung der bisher realisierten Konzepte von MILOS erfolgen. Es folgt eine kurze Übersicht der resultierenden Arbeitspakete.

Konzeption eines Wissensmanagement-Systems

Für die angestrebte Realisierung des Wissensmanagements in MILOS sind die folgenden wissenschaftlichen Fragen von Interesse: (a) Wie kann der Projektzustand bei

Entscheidungen während der Projektausführung zum Auffinden relevanten Wissens berücksichtigt werden? (b) Wie können Informationsbedürfnisse modelliert werden, die dem Wissensmanagement eine pro-aktive Arbeitsweise ermöglichen? (c) Wie können Aktivitäten zur Pflege der Wissensquellen und der modellierten Informationsbedürfnisse in den abzuwickelnden Prozess integriert werden? (d) Inwiefern können Ontologien, Terminologien und Thesauri das Wissen strukturieren und somit dem Wissensmanagement zu mehr Nutzen verhelfen?

Da Entscheidungen letztendlich immer vom zuständigen Benutzer zu treffen sind und nicht für Inferenzen verwendet werden, sollen zur Lösung genannter Fragen Assistenzprozesse spezifiziert werden, welche Retrieval, Pflege und Wartung von Wissen unterstützen und ihrerseits selbst ereignisgesteuert durch MILOS abgearbeitet werden. Hierdurch ist es möglich, auch komplexe Prozesse des Wissensmanagements, die den Eingriff des Benutzers erfordern, flexibel zu gestalten und zu dokumentieren. Der gesamte Prozess kann somit als Kooperationsprozess von Wissen- und Projektmanagement aufgefasst werden.

Integration heterogener Wissensquellen

Das zugrundeliegende Wissen soll schwerpunktmäßig aus der in Teilprojekt A1 entwickelten und in Teilprojekt A3 implementierten Erfahrungsdatenbank (SFB-EB) stammen sowie auch andere Wissensquellen (z. B. existierende Ressourcen, Dokumenten- bzw. Data-Management-Systeme) umfassen können. Das begriffliche Hintergrundwissen über Inhalte der Wissensquellen ist für die hier intendierten Zwecke noch nicht zusammenhängend dargestellt und soll daher in Form von Ontologien, Terminologien und Thesauri im Rahmen dieses Teilprojekts erfolgen.

Die hiermit angestrebte Form der Assistenz soll als Prozess des Wissensmanagements ebenfalls durch MILOS abgewickelt werden. Hierzu sollen vorab für jede Aktivität Agenten zur Informationssuche definiert werden können. Während der Projektdurchführung stehen die Suchergebnisse somit den jeweiligen Bearbeitern einer Aufgabe unmittelbar zur Verfügung bzw. führen zu einer Benachrichtigung über ihr Vorhandensein. Durch diese enge Verzahnung des Zugriffs auf die Wissensquellen in den Prozessablauf wird eine situationsspezifische Behandlung erst möglich.

Unterstützung bei der Anpassung der Messinstrumentierung auf den laufenden Entwicklungsprozess

Basierend auf den im zuvor genannten Arbeitsschwerpunkt zu entwickelnden Grundlagen soll ein konkreter Assistenzprozess spezifiziert und integriert werden, der den Benutzer bei der Anpassung der Messinstrumentierung unterstützt. Diese Anpassung ist zur Erhaltung der Konsistenz zwischen Projekt- und Messplan notwendig und muss zusammen mit Planänderungen bzw. -verfeinerungen vorgenommen werden. Ziel dieses Arbeitsschwerpunktes ist, die Nützlichkeit der Berücksichtigung ontologischen und terminologischen Wissens sowie die pro-aktive Vorgehensweise in Verbindung mit Erfahrungswissen exemplarisch anhand eines hochgradig wissensintensiven Prozesses zu demonstrieren. Ergebnis dieses Arbeitsschwerpunktes soll eine deutliche

Steigerung der Berücksichtigung von Wiederverwendungs-Artefakten (z.B. Messpläne) sein, die im Rahmen kontrollierter Experimente im SE Labor zu ermitteln ist.

Bereitstellung bewerteter Projektzustände

Die Verfeinerung des Projektplans während der Projektausführung und die zielgerichtete Reaktion auf Abweichungen von Projektzielen bedarf einer systematischen Kontrolle. Der in Teilprojekt B1 zu entwickelnde Prozessleitstand visualisiert den aktuellen Projektzustand und interpretiert Messdaten unter Zuhilfenahme von Vergleichswerten aus vergangenen Projekten. Somit stehen für die Qualitätssicherung sowie das Projektmanagement aufbereitete und teilweise interpretierte Informationen bereit, die als Grundlage für Entscheidungen genutzt werden können. Hierdurch wird eine Verbesserung der Sichtbarkeit des Projektzustands erreicht, auf deren Basis Vorschläge zur zielgerichteten Reaktion durch das Projektmanagement generiert werden können. Dadurch kann folgendes erreicht werden: (a) Verbesserung der Entscheidungsgrundlage für Umplanungen; es stehen zusätzlich zu der allgemeinen Projektcharakterisierung detailliertere Informationen bereit, mit deren Hilfe Ursachen für Planabweichungen beziehungsweise für die Nichteinhaltung von Qualitätszielen erkannt werden können. (b) Erlangung einer größeren Übereinstimmung zwischen Projektplan und realem Entwicklungsprozess durch frühzeitige Detektion von Abweichungen und kritischen Projektzuständen. (c) Verstärkung der Projektkontrolle. (d) Verbesserte Wiederverwendung von Erfahrungswissen während der Projektausführung durch genauere Kenntnis des Wiederverwendungskontexts.

Der Prozessleitstand stellt damit ein geeignetes Werkzeug dar, die Wiederverwendung unter Berücksichtigung des Projektzustands zu unterstützen und hilft darüberhinaus dem Projektmanagement und der Qualitätssicherung bei der Projektkontrolle und -steuerung. Er soll aus diesen Gründen an MILOS angebunden werden.

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Grundlagen zur Integration von Methoden des Wissensmanagements und MILOS

In diesem Arbeitsschwerpunkt sollen Grundlagen als Basis für Wissensmanagement in MILOS erforscht werden. Hierzu zählen Konzepte bzgl. der Modellierung von Informationsbedürfnissen, der zielgerichteten Reaktion auf Ereignisse von Seiten des Wissensmanagements, der Strukturierung von Wissen mit Hilfe von Ontologien, Terminologien und Thesauri. Ausserdem ist eine Architektur zur Einbettung autonomer Computeragenten zu entwerfen, welche die technische Basis für die geplante proaktive Unterstützung in Form von Assistenten darstellt.

Arbeitspaket A1: Identifikation und Klassifikation von Informationsbedürfnissen

Die Analyse der bisherigen Planung und Umplanung von Team-Entwicklungsprozessen im Hinblick auf den Informationsbedarf (z. B. Wissen über den aktuellen Stand von Ressourcen wie Fachkompetenzen von Mitarbeitern oder Wissen über Gebäude-

automation) ist eine wichtige Vorarbeit und soll im Rahmen dieses Arbeitspakets vorgenommen werden. Das Ergebnis dieser Analyse soll als Basis für die Modellierung von Informationsbedürfnissen herangezogen werden, die in Zusammenarbeit mit anderen Teilprojekten, vor allem Teilprojekt B2, geeignet kategorisiert werden. Kategorisieren bedeutet in diesem Zusammenhang einen Bezug zwischen den Informationsbedürfnissen und dem Projektzustand herzustellen, in dem sie relevant sind. Die Informationsbedürfnisse bilden die Grundlage der pro-aktiven Unterstützung des Wissensmanagements.

Personalbedarf: 9 PM

Arbeitspaket A2: Verwaltung von Abhängigkeiten der Informationsbeschaffung

In diesem Arbeitspaket sollen unter Einbezug anderer Teilprojekte (z. B. Teilprojekt B2) Methoden und Techniken entwickelt werden, Benutzer pro-aktiv über Möglichkeiten zur Entscheidungsverbesserung informieren zu können. Hierzu ist es einerseits notwendig, aus den modellierten Informationsbedürfnissen geeignete Fragen zu generieren, andererseits entsprechende Triggerbedingungen spezifizieren zu können, die einen Wissensmanagementprozess (Assistenzprozess zur Beantwortung von Fragen, Behebung von Inkonsistenzen etc.) anstoßen. Z. B. könnte eine Änderung am Bestand der Ressourcen (Resourcepool) zuvor gelieferte Resultate von Suchanfragen ungültig machen und für bereits auf Basis dieser Resultate getroffene Entscheidungen von Bedeutung sein. Des weiteren sollen Mechanismen zur Verwaltung von Informationswünschen im Fall einer sich ändernder Informationslage zur Verfügung gestellt werden. Durch eine zu realisierende automatische Überwachung der Resultate von Suchanfragen kann z. B. eine automatische Benachrichtigung erfolgen, wenn geeignetere Resultate vorliegen.

Personalbedarf: 8 PM

Arbeitspaket A3: Spezifikation von Ontologien und Aufbau von Thesauri

Domänenspezifische Ontologien, Terminologien und Thesauri ermöglichen unter Berücksichtigung von Inferenzschritten (z. B. mittels Spezialisierungsrelation) zum einen die Suche in heterogenen Wissensquellen sowie die anschließende Verarbeitung der Ergebnisse. Zum anderen lassen sich hierüber im Falle zunächst erfolgloser Suchanfragen geeignete Umformulierungen der Fragestellung extrahieren. In diesem Arbeitspaket sollen zunächst Anforderungen an die Formalisierungssprache untersucht werden, auf deren Basis umfangreiche unternehmens- und domänenspezifische Ontologien erstellt werden können. Dies soll unter Berücksichtigung der Integration mit Techniken des fallbasierten Schließens geschehen, welche bereits von der SFB-EB angeboten werden.

Da das begriffliche Wissen über die SE-Domäne im allgemeinen und die SFB-Anwendungsdomäne (Gebäudeautomation) im besonderen noch nicht vollständig dargestellt ist, sind hier die von den Teams durchgeführten SE-Prozesse sowie das im SFB entwickelte Domänenmodell für die Gebäudeautomation zu analysieren und zu

konzeptualisieren, sowie Strategien zu deren Verwaltung und Anpassung zu entwerfen.

Personalbedarf: 6 PM

Arbeitspaket A4: Entwicklung eines Agenten-Frameworks zur Informationssuche

In diesem Arbeitspaket soll ein Agenten-Framework in MILOS integriert werden, das die Entwicklung von autonomen Computeragenten zur flexiblen Informationssuche in heterogenen Wissensquellen auf Basis der erstellten Ontologien erlaubt. Es werden beispielhaft Agenten entwickelt, die auf vorhandene Wissensquellen zugreifen und eine situationsspezifische Suche nach relevanten Planungs- bzw. Ausführungsinformationen durchführen. Hierzu gehört auch der Zugriff auf frühere Projektablaufe (Traces), für den in Zusammenarbeit mit dem Teilprojekt B2 geeignete Repräsentationstechniken für Projekt ereignis- und Entscheidungsketten sowie begriffliche Indizierungsschemata entwickelt werden.

Personalbedarf: 24 PM

Arbeitsschwerpunkt B: Unterstützung der Anpassung der Messinstrumentierung bei Umplanung

Dieser Arbeitsschwerpunkt soll exemplarisch eine Verfeinerung der in Arbeitsschwerpunkt A zu erarbeitenden Konzepte behandeln, um einerseits deren Nützlichkeit zu demonstrieren und andererseits Methodiken der Spezifikation komplexer Assistenzprozesse für das Projektmanagement aufzuzeigen. Er definiert die Schnittstelle zum Prozessleitstand, der einerseits für die Interpretation von Messergebnissen zuständig ist, andererseits konzeptionelles Wissen über Qualitätsmodelle als Hintergrundwissen für Messpläne zur Verfügung stellt.

Arbeitspaket B1: Identifikation einer Methodik zur Assistenz und Pflege des Hintergrundwissens

Da die Anpassung der Messinstrumentierung über ein einstufiges kontextbasiertes Retrieval hinausgeht und des weiteren die Einbindung des Benutzers in den Entscheidungsprozess notwendig ist, soll in diesem Arbeitspaket zunächst eine sinnvolle Methodik bei der Assistenz und Verwendung von Erfahrungswissen speziell für diese Problematik in Zusammenarbeit mit den Teilprojekten B1 und B2 ermittelt werden. Diese Methodik stellt in Verbindung mit dem vom Prozessleitstand zur Verfügung gestellten konzeptionellen Wissen über Messinstrumentierung in Form von Qualitätsmodellen und Wissen über deren Anwendung eine Ontologie für das zu erstellende Assistenzsystem dar.

Personalbedarf: 8 PM

Arbeitspaket B2: Modellierung des Assistenzprozesses in MILOS

Die Akzeptanz von Computerassistenten durch den Benutzer hängt u. a. davon ab, in wie weit Benutzervorgehen und Assistenz übereinstimmen. Da diese notwendige Eigenschaft bei fest vorgegebenem Assistentenverhalten nicht für jeden Benutzer

garantiert werden kann, bietet sich eine Modellierung und Ausführung des Assistenzsystems als (Meta-)Prozess an, um die von MILOS zur Verfügung gestellte Möglichkeit der Umplanung während der Ausführung auch auf das Assistenzsystem zu übertragen und somit die nötige Flexibilität zu erreichen. Das Resultat ist eine ineinander verzahnte Ausführung des Metaprozesses der Planung und des Wissensmanagementprozesses zur Assistenz. Diese Vorgehensweise verspricht des weiteren, Merkmale, die sich aus der verteilten Planung bzw. Ausführung ergeben, wie Einschränkung des Zugriffs auf Wissensquellen etc., zu integrieren.

Personalbedarf: 4 PM

Arbeitspaket B3: Integration notwendiger Schritte zur Pflege von Ontologien in den Assistenzprozess

Die in Arbeitspaket B2 beschriebene Vorgehensweise erfordert eine ständige Pflege von Hintergrundwissen (Ontologien), z. B. in Form von Prozessmodellen des Assistenzsystems. Des weiteren soll auch die Integration neuer Qualitätsmodelle bzw. Wissen über deren Anwendung möglich sein. Die hierfür notwendigen elementaren Schritte, wie Konsistenzchecks, Neudefinition und Revision von Konzepten etc., sollen im Rahmen dieses Arbeitspakets ebenfalls identifiziert und in das Prozessmodell des Assistenzsystems aufgenommen werden.

Personalbedarf: 8 PM

Arbeitspaket B4: Entwicklung von Agenten zur (teil-) automatischen Abarbeitung des Assistenzprozesses

Die Modellierung des Assistenzprozesses zur Anpassung der Messinstrumentierung als MILOS Metaprozess stellt eine Möglichkeit der Dokumentation dar. Weiterhin lässt sie die Möglichkeit einer genauen und flexiblen Spezifikation des Zusammenspiels von Mensch und Maschine, indem einzelne Tasks wahlweise Personen oder Computeragenten zugeordnet werden können. In diesem Arbeitspaket sollen Agenten zur Abwicklung in Arbeitspaket B2 spezifizierter Prozesse entworfen werden, die auf Basis des in Arbeitspaket A4 zu konzipierenden Agenten-Frameworks einzelne Tasks ausführen (Retrieval, Konsistenzchecks etc.) und somit die erwünschte Computerassistenz während der Umplanung realisieren.

Personalbedarf: 13 PM

Arbeitsschwerpunkt C: Technische Anbindung des Prozessleitstands

In diesem Arbeitsschwerpunkt soll der in Teilprojekt B1 zu entwickelnde Prozessleitstand in das MILOS-System integriert werden. Um eine solche Integration zu gewährleisten, sind im MILOS-System die technischen Voraussetzungen zu schaffen, die sowohl die Messinstrumentierung als auch die Umplanungsunterstützung (insbesondere die Präsentation und Weiterverarbeitung von entscheidungsrelevanten Informationen) betreffen. Zusätzlich ist auf Basis der in MILOS zu entwickelnden Methodik des Wissensmanagements die explizite Verwaltung von Qualitätsmodellen vorgesehen. Diese stellen notwendiges Hintergrundwissen für den Prozessleitstand dar.

Arbeitspaket C1: Voraussetzungen für den Prozessleitstand

Gemäß der Konstrukte der MILOS-Sprache können Produkte, Prozesse und Ressourcen zum Zwecke der Messdatenerfassung attribuiert werden. Die Ausführungsmaschine soll demgemäß Mechanismen beinhalten, um diese Messdaten manuell, semi-automatisch bzw. vollautomatisch zu erfassen und auf geeignete Weise bereitzustellen, so dass einerseits die Projektsteuerung (insbesondere das Weiterschalten des Projektzustands) als auch der Prozessleitstand hierauf zugreifen können. Die Erfassung der Messdaten kann durch die Anbindung von Messwerkzeugen an MILOS zusätzlich unterstützt werden.

Personalbedarf: 19 PM

Arbeitspaket C2: Einbindung von Qualitätsmodellen

Qualitätsmodelle repräsentieren Erfahrungswissen bzgl. relevanter Qualitätseigenschaften von Projekten, Prozessen, Produkten und Ressourcen. Sie werden im Rahmen von Fallstudien und kontrollierten Experimenten gewonnen, von Teilprojekt B1 aufbereitet und über die SFB-EB zur Verfügung gestellt. Die technische Integration von Qualitätsmodellen in MILOS erfordert eine Erweiterung der Prozessnotation sowie die Abstimmung auf Formalisierungstechniken des Wissensmanagements.

Personalbedarf: 9 PM

Arbeitspaket C3: Visualisierung der Messdaten

Die durch die Interpretation der Messdaten gewonnen Informationen müssen dem Planer adäquat zur Verfügung gestellt werden. Dies erfordert die Implementierung eines Planungsfensters für den Projektmanager, in dem die entscheidungsrelevanten Informationen bzgl. Umplanungen oder Auswahl von Wiederverwendungsartefakten präsentiert werden.

Personalbedarf: 4 PM

Arbeitsschwerpunkt D: Evaluierung entwickelter Techniken und Werkzeuge

In diesem Arbeitsschwerpunkt sollen die neu hinzugekommenen Komponenten von MILOS im Rahmen von Fallstudien und formalen Experimenten erprobt werden. Zu untersuchende Hypothesen betreffen insbesondere die Reduktion des Aufwands und die Bedienbarkeit. Bei der Evaluierung wird die vom Teilprojekt B1 bereitgestellte Experimentiermethodik angewendet.

Arbeitspaket D1: Integration ins Software-Engineering-Labor

Im Rahmen des Arbeitspakets D1 soll MILOS in Zusammenarbeit mit Teilprojekt A1 im SE-Labor zur Verfügung gestellt werden. Hierdurch werden andere Teilprojekte in die Lage versetzt, ihre Software-Entwicklungen mit Hilfe von MILOS durchzuführen. Dies bedarf der Schulung der Mitarbeiter der anderen Teilprojekte über die Konzepte und Benutzung von MILOS. Zusätzlich sollen im Rahmen dieses Arbeitspakets Anforderungen anderer Teilprojekte in MILOS umgesetzt werden.

Personalbedarf: 28 PM

Arbeitspaket D2: Durchführung von Fallstudien

Nachdem MILOS im SE-Labor zur Verfügung steht, sollen in diesem Arbeitspaket Fallstudien im Rahmen von Applikationsentwicklungen des SFB durchgeführt werden. Im Vordergrund der Fallstudien stehen die Anleitung von management- und technisch-orientierten Rollen, die Unterstützung von Umplanungsaktivitäten sowie die automatisierte Erfassung von Messdaten während der Projektausführung.

Personalbedarf: 9 PM

Arbeitspaket D3: Durchführung kontrollierter Experimente

In diesem Arbeitspaket werden experimentell Aspekte bezüglich des Nutzens der neu hinzugekommenen Elemente des Werkzeugs für management- und technisch-orientierte Rollen in kontrollierten Experimenten untersucht. Dabei sollen besonders die in dieser Förderperiode neu hinzukommenden Bestandteile von MILOS experimentell erprobt werden. Beispielsweise kann der Aufwand zur Umplanung der Messinstrumentierung untersucht werden, der durch die Änderung einer Anforderung verursacht wird. Eine mögliche Hypothese ist, dass sich mit Unterstützung des in diesem Teilprojekt entwickelten Werkzeugs dieser Aufwand signifikant gegenüber der Verwendung von MILOS ohne die neuen Komponenten reduziert.

Personalbedarf: 4 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausrüstung						
Schmitt, S.(AG Richter)		B1				
				B2	B3	
	A4					
N.N. (AG Rombach)		C2				
			C1		A3	
Ergänzungsausrüstung						
Petersen, E. (AG Rombach)	A1		A2		C3	
			D1			
			D2		D3	
Schaaf, M. (AG Richter)	A2		A3		B4	
	C1	A1		A4		
				B2	B3	
N.N., stud. Hilfskraft			C1			
					B4	
N.N. stud. Hilfskraft			D1			
					C3	
N.N. stud. Hilfskraft		C2				
			A4			

Im Zeitplan ist ein Aufwand von wenigstens 8 Std. / Woche für SFB-übergreifende Themenstellungen berücksichtigt, die jedes Teilprojekt zur Verfügung stellt. Diese Arbeiten werden in Teams organisiert und im Arbeitsbericht gesondert beschrieben.

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Das in der vergangenen Förderungsperiode erstellte Werkzeug MILOS wird durch die Integration mit Teilprojekt A1 ins SE-Labor allen anderen Teilbereichen für Ihre Software-Entwicklungen zur Verfügung gestellt.

Der im Teilprojekt B1 entwickelte Prozessleitstand wird an MILOS angebunden, um den Projektmanager bei der Ermittlung des Projektzustands zu unterstützen.

Zusammen mit Teilprojekt A1 und A3 wird die Anbindung von MILOS an die SFB-Erfahrungsdatenbank weitergeführt.

Teilprojekt B2 beschäftigt sich mit der Problematik der Meta-Modellierung von Planungsprozessen. Dort identifizierte Abhängigkeiten der Planung untereinander bzw. Maßnahmen zur Erhaltung der Plankonsistenz liefern interessante Hinweise für eine pro-aktive wissensbasierte Assistenz durch das Teilprojekt A2. Des weiteren kann Teilprojekt B2 bei der Formalisierung von Planungswissen (Konsistenzbedingungen etc.) und dessen Verwaltung durch Teilprojekt A2 unterstützt werden.

Die Evaluierung des Werkzeug-Prototypen erfolgt entsprechend der von Teilprojekt B1 definierten Experimentiermethodik.

Mit den Teilprojekten A1 und D1 ist eine Zusammenarbeit bei der Durchführung und Auswertung der Experimente aus Arbeitsschwerpunkt D vorgesehen.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	1	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
			StdHkr	3	61.200	StdHkr	3	61.200	StdHkr	3	61.200
	zus.:	1	zus.:	5	270.000	zus.:	5	270.000	zus.:	5	270.000
SV			Kostenkategorie oder Kennziffer	Betrag in DM		Kostenkategorie oder Kennziffer	Betrag in DM		Kostenkategorie oder Kennziffer	Betrag in DM	
			(515)	-	(515)	-	(515)	-			
			(522)	3.000	(522)	3.000	(522)	3.000			
			zusammen	3.000	zusammen	3.000	zusammen	3.000			
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Richter, M. M., Prof. Dr. (2) Rombach, H. D., Prof. Dr. (3) Schmitt, S., Dipl-Inform. (AG Richter) (4) 1/2 N.N., Dipl-Inform. (AG Rombach)	Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	5-10 5-10 20 10	1.1.95 1.1.95 1.1.01 1.1.01	
Ergänzungsausstattung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Petersen, E., Dipl-Inform. (AG Rombach) (2) Schaaf, M., Dipl-Inform. (AG Richter) (3) N.N., StdHkr (4) N.N., StdHkr (5) N.N., StdHkr	Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	38,5 38,5 19 19 19	2.9.99 15.2.99 1.1.01 1.1.01 1.1.01	BAT IIa BAT IIa StdHkr StdHkr StdHkr
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Der erste wissenschaftliche Mitarbeiter (AG Richter) identifiziert grundlegende Methodiken zur wissensbasierten Assistenz und beteiligt sich an deren Realisation auf Basis eines Agenten-Frameworks.

Der zweite wissenschaftliche Mitarbeiter (AG Rombach) bindet Qualitätsmodelle in MILOS ein und beteiligt sich an deren Formalisierung hinsichtlich der gewählten ontologischen Beschreibungssprache. Des weiteren wirkt er bei der Anbindung des Prozessleitstands mit.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Die erste wissenschaftliche Mitarbeiterstelle (AG Rombach) erarbeitet konzeptionelle Grundlagen für Arbeitsschwerpunkt B. Des weiteren ist er verantwortlich für die Einführung des MILOS Systems in das SE-Labor und dessen systematische Evaluation.

Der zweite wissenschaftliche Mitarbeiter (AG Richter) führt die zentralen Arbeiten zur Integration von Methoden des Wissensmanagements und zur Unterstützung der Anpassung der Messinstrumentierung aus. Eine weitere Aufgabe ist die Sicherstellung der technischen Voraussetzung zur Anbindung des Prozessleitstands.

Die erste studentische Hilfskraft hilft zunächst bei den Implementierungsarbeiten zur Anbindung des Prozessleitstands (Arbeitspaket C1) und unterstützt bei der Entwicklung von Agenten zur automatischen Abarbeitung von Assistenzprozessen (Arbeitspaket B4).

Die zweite studentische Hilfskraft unterstützt die Integration von MILOS in das Software-Engineering-Labor (Arbeitspaket D1) und hilft im weiteren Verlauf des Teilprojekts bei der Visualisierung von Messdaten (Arbeitspaket C3).

Die dritte studentische Hilfskraft hilft bei der Implementierung des in Arbeitspaket A4 zu konzipierenden Agenten-Frameworks sowie bei der Integration von Qualitätsmodellen in MILOS (Arbeitspaket C2).

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausstattung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausstattung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausstattung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausstattung der sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausstattung werden zur Finanzierung von Literaturrecherchen und elektronischem Kleinmaterial benötigt. Ausserdem sollen die in der vergangenen Förderungsperiode zur Entwicklungszwecken angeschafften PCs schrittweise erweitert werden. Die jetzigen Geräte sind für die effektive Arbeit mit dem in GemStone eingebetteten MILOS System und den aktuellen integrierten Java Entwicklungsplattformen nicht mehr ausreichend.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
-	-	-	-

3.1 Allgemeine Angaben zum Teilprojekt A3

3.1.1 Thema

Unterstützung des Softwareentwicklungsprozesses durch objekt-relationale Datenbank-Technologie

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Datenbanken und Informationssysteme

3.1.3 Leiter

Prof. Dr. Theo Härder
geb. 28.08.1945
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-4030
Fax: 0631/205-3299
haerder@informatik.uni-kl.de

Dr. Norbert Ritter
geb. 07.10.1962
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-3264
Fax: 0631/205-3299
ritter@informatik.uni-kl.de

Die C1-Stelle von Dr. Ritter ist befristet bis zum 28.02.2001 mit der Verlängerungsmöglichkeit von 3 Jahren.

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ ja	☒ nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ ja	☒ nein
Tierversuche	☐ ja	☒ nein
gentechnologische Untersuchungen	☐ ja	☒ nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
1998	201,6	13,0	-	214,6
1999	206,4	3,0	-	209,4
2000	208,8	3,0	-	211,8
Zwischensumme	616,8	19	-	635,8
2001	270,0	3,0	-	273,0
2002	270,0	3,0	-	273,0
2003	270,0	3,0	-	273,0

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Wegen ihrer Flexibilität und Erweiterbarkeit bietet sich die Nutzung objekt-relationaler Datenbanktechnologie im Bereich der Softwareentwicklung an, da sowohl der eigentliche Softwareentwicklungsprozess als auch die zu entwickelnden Produkte (Softwaresysteme) Datenverwaltungs-komponenten benötigen, die verschiedenartigste Datentypen und Verarbeitungscharakteristika zu unterstützen in der Lage sind. Dieses Teilprojekt hat sich zum primären Ziel gesetzt, Konzepte für die Nutzung objekt-relationaler DB-Technologie in beiden angesprochenen Bereichen zu erarbeiten. Weiterhin will es dazu beitragen, die künftige Entwicklung objekt-relationaler DBVS (ORDBVS), ausgehend von der SFB-Domäne, durch neue Konzepte und Mechanismen voranzutreiben.

In der vergangenen Förderungsperiode sind

- Erweiterungsmechanismen untersucht und evaluiert worden sowie
- ein Prototyp eines Benchmarks für ORDBVS,
- Basiskonzepte für die Handhabung multimedialer Daten in ORDBVS,
- grundlegende Vorgehensweisen für die generische Entwicklung von ORDB-Anwendungen,
- ein flexibles Verarbeitungsmodell für ORDBVS,
- ein Prototyp einer ORDB-gestützten Erfahrungsdatenbank

entwickelt worden. Aufbauend auf diesen Arbeiten sollen in der kommenden Förderungsperiode neue Schwerpunkte gesetzt werden, wobei

- die Nutzung von OR-Technologie zur Entwicklung eines umfassenden SE-Repository ausgehend von dem bereits realisierten Prototyp der SFB-501-Erfahrungsdatenbank (SFB-501-EDB),

- die methodische Vorgehensweise bei der Entwicklung von ORDB-Anwendungen als integrale Bestandteile der in SE-Prozessen entwickelten Produkte (Softwaresysteme),
- begleitende Untersuchungen objekt-relationaler Technologie

im Vordergrund stehen.

Ausgehend von dem gegenwärtigen ORDB-gestützten Prototyp der SFB-501-EDB soll in dem ersten der drei genannten Arbeitsschwerpunkte ein umfassendes SE-Repository angestrebt werden, das eine Integration von Erfahrungsdaten- und Produktdatenverwaltung sowie der Ablaufkontrolle leistet. Auf diese Weise wird ein integriertes, mit einer einheitlichen Schnittstelle ausgestattetes System bereitgestellt, das den gesamten Softwareentwicklungsprozess umfassend unterstützt. Insbesondere wegen ihrer Modellierungskonzepte und ihrer Mechanismen zur Erweiterbarkeit bietet sich die ORDBVS-Technologie als Vehikel für eine solche Integration an. Dennoch sei bereits an dieser Stelle betont, dass es hier weniger um die vollständige Neuentwicklung eines umfassenden SE-Repository geht, als vielmehr um die Ausgestaltung vorhandener Komponenten (wie z. B. der Erfahrungsdatenbank und der Ablaufsteuerungskomponente MILOS) und die Ermöglichung einer Interoperabilität dieser Komponenten unterhalb einer gemeinsamen Repository-Schnittstelle.

Der zweite Arbeitsschwerpunkt soll sich mit der Entwicklung von ORDB-Anwendungen mit generischen Methoden beschäftigen. Letztere werden zur Entwicklung von ORDB-basierten Datenverwaltungskomponenten genutzt, die wiederum als Kernkomponenten großer Softwaresysteme betrachtet werden können. Die besondere Herausforderung besteht hier darin, Konzepte und Verfahren zu entwickeln, die eine angemessene Ausnutzung der Erweiterbarkeitsmechanismen erlauben. Dabei ist zu entscheiden, welche der benötigten Datenverwaltungsfunktionen als Erweiterungen des ORDBVS und welche oberhalb der DBVS-Schnittstelle (Application-Server) realisiert werden sollten. Weiterhin soll untersucht werden, wie sich diese Komponenten generisch bereitstellen lassen.

Der dritte Arbeitsschwerpunkt beschäftigt sich mit eher grundlegenden Aspekten objekt-relationaler Datenbanktechnologie und dient dazu, die allgemeine Entwicklung und Gestaltung des objekt-relationalen Datenmodells und zugehöriger Systemkomponenten im Auge zu behalten bzw. durch eigene Beiträge zu beeinflussen. Insbesondere die Erweiterbarkeit (Tiefenintegration), der Internet-Zugang und die Leistungsbewertung von ORDBVS sind hier zu nennen. Entsprechende Untersuchungen erleichtern auf der einen Seite die Nutzung vorhandener OR-Konzepte im Bereich der Softwareentwicklung und auf der anderen Seite können die Ergebnisse dieser Untersuchungen dazu genutzt werden, die Entwicklung von ORDBVS dahingehend zu beeinflussen, dass die Anforderungen der SE-Domäne angemessen berücksichtigt werden.

3.3 Stand der Forschung

Obwohl ORDBVS [5] mittlerweile mit dem SQL:99-Standard [6, 7] hinsichtlich ihrer Schnittstelle standardisiert worden sind und auch erste ORDBVS als Weiterentwicklungen erfolgreicher RDBVS kommerziell verfügbar sind, liegen nur sehr rudimentäre Erkenntnisse darüber vor, wie angesichts spezieller Anwendungscharakteristika die Erweiterbarkeitseigenschaft dieser Systeme optimal genutzt werden kann. Dies bedeutet, dass konkretes und detailliertes Know-how fehlt, wie neue Anwendungen hinsichtlich der Nutzung der Erweiterbarkeit sinnvoll zu implementieren sind. Darüber hinaus ist unklar, wie geeignete DBVS-Mechanismen zur Erweiterbarkeit gestaltet und implementiert werden sollten.

Hinsichtlich einer gezielten Unterstützung von Softwareentwicklungsanwendungen durch ORDBVS kennen wir neben unserem Teilprojekt A3 des SFB 501 zur Zeit keine weiteren Forschungsaktivitäten. Obwohl die Speicherung und Verwaltung von Software-Artefakten bereits seit etwa zwei Jahrzehnten in der Forschung betrachtet wird [4], wurden bisher im wesentlichen maßgeschneiderte Systeme bzw. auf relationalen oder objektorientierten DBVS basierende Systeme vorgeschlagen. Beispiele für andere Ansätze von Erfahrungsdatenbanken sind in [1, 3, 8] beschrieben. Neben der Tatsache, dass diese Systeme nicht auf ORDBVS aufbauen, unterscheiden sie sich vom SFB-501-EDB-Ansatz insbesondere darin, dass der wesentliche Schwerpunkt auf der Erfassung und Bereitstellung von Erfahrungsdaten liegt und keine integrierte Verwaltung von Erfahrungs-, Produkt- und Ablaufkontrolldaten beabsichtigt ist. Wir halten eine solche Integration jedoch für sehr sinnvoll, da nur so eine geeignete Verknüpfung verschiedener Arten von Daten erreicht werden kann. Auf diese Weise können Messdaten geeignet zugeordnet und die Analyse von experimentspezifischen Daten vereinfacht werden, sodass diese Experimente einfacher nachvollzogen werden können. Die Nutzung von ORDBVS als Integrationsvehikel bietet sich an, da diese entsprechende Modellierungsmächtigkeit und Erweiterungsmöglichkeiten bieten und in einfacher Weise an das WWW angebunden werden können. In [9] wird ebenfalls ein integrierter Ansatz einer Entwurfsumgebung beschrieben, jedoch wird hier keine objekt-relationale DB-Technologie betrachtet.

Dennoch erfüllen auch ORDBVS nicht alle sich ergebenden Anforderungen. Dies gilt insbesondere für die unscharfe und iterative Suche bei neuen Datentypen. Beispielsweise haben sich hauptspeicherbasierte, Methoden des Case-Based-Reasoning verwendende Werkzeuge zur ähnlichkeitsbasierten Suche nach Erfahrungsdaten [4], allerdings im Einbenutzerbetrieb, als sehr hilfreich und effizient erwiesen. Inwieweit eine allgemeinere, ORDBVS-basierte Lösung ähnlich effizient sein kann bzw. durch welche (möglicherweise neuen) Erweiterungsmechanismen sie ähnlich effizient gestaltet werden kann, soll Gegenstand der Forschung der kommenden Periode sein.

Ein weiteres Defizit von ORDBVS hinsichtlich der Unterstützung von Entwurfsanwendungen liegt im ausschließlich Server-zentrierten Verarbeitungskonzept. Entwurfsanwendungen benötigen Komponenten zur Pufferung und Verarbeitung der DB-Daten auf dem Client. Konzeptionell gibt es hier eine Reihe von Verarbeitungsmodel-

len, die sogar ein Nebeneinander von Server- und Client-basierter Verarbeitung erlauben [2]. Hier ist es jedoch notwendig, einer Anwendungsfunktion bereits bei ihrer Registrierung einen Ausführungsort statisch zuzuordnen. Um die Dynamik von Entwurfsabläufen jedoch geeignet unterstützen zu können, halten wir ein flexibleres Verarbeitungskonzept für notwendig, das dynamisch zu entscheiden in der Lage ist, ob eine Funktion auf der Client- oder Server-Seite effizienter ausgeführt werden kann. Ein bereits vorliegendes Konzept soll in der kommenden Förderungsperiode verfeinert und evaluiert werden und es sollen Möglichkeiten aufgezeigt werden, wie ein ORDBVS mit einem solchen flexiblen Verarbeitungskonzept ausgestattet werden kann.

Weiterhin unterstützen ORDBVS gegenwärtig keine Versionierung. Die Möglichkeiten der datenmodellbezogenen Erweiterungen (benutzerdefinierte Typen und Funktionen) sind hinsichtlich der Implementierung von Versionierungsfunktionalität eher ungeeignet, da ein Versionsmodell in der Regel ein eigenes Datenmodell darstellt und sich daher nur sehr schlecht in die SQL:99-Umgebung einbetten lässt. Daraus folgt, dass entsprechende Application-Server-Architekturen eingeführt werden müssen, welche die eigentliche Versionierungsfunktionalität an der Application-Server-Schnittstelle anbieten. Dennoch können natürlich auch bei diesem Ansatz Basisfunktionen der Versionierung als ORDBS-Erweiterungen realisiert werden.

Das Beispiel der Versionierung verdeutlicht auch, dass es sinnvoll ist, die Entwicklung von ORDB-Anwendungen durch generische Methoden zu unterstützen. Da es sehr viele unterschiedliche Versionierungsmechanismen [9] gibt und sich verschiedene Anwendungen in ihren Anforderungen diesbezüglich unterscheiden, verfolgen wir den Ansatz, ein Basis-Versionierungs-Framework anzubieten, auf dem aufbauend spezifische Anwendungsanforderungen in abstrakter Form (UML) spezifiziert werden können. Die anschließende Anwendungsentwicklung soll dann generisch erfolgen. Die bisher in keinen anderen Forschungsprojekten betrachtete Zielsetzung bei dieser Vorgehensweise ist es, automatisch zu entscheiden, welche Teile der Anwendungsfunktionalität als ORDBVS-Erweiterung und welche als Application-Server-Funktionen realisiert werden sollten. Diese vollkommen neue Vorgehensweise der generischen ORDB-Anwendungsentwicklung soll natürlich nicht auf die Entwicklung von Versionierungsfunktionalität beschränkt werden, sondern auch für andere Arten von ORDB-Anwendungen genutzt werden können.

Literatur

- [1] Althoff, K.-D., Birk, A., Hartkopf, S., Müller, W., Nick, M., Surmann, D., Tautz, C.: *Managing Software Engineering Experience for Comprehensive Reuse*, Technischer Bericht 001.99, Fraunhofer Institut für Experimentelles Software-Engineering, 1999.
- [2] Franklin, M.J., Carey, M.J., Livny, M.: *Transactional Client-Server Cache Consistency: Alternatives and Performance*, ACM TODS 22:3, 1997, 315-363.
- [3] Henninger, S.: *An Environment for Reusing Software Processes*, Proc. 5th. IEEE Int. Conf. on Software Reuse (ICSR5), Victoria, BC, Kanada, 1998.

- [4] Mili, A., Mili, R., Mittermeir, R.T.: *A survey of software reuse libraries*, Annals of Software Engineering 5, 1998, 349-141.
- [5] Stonebraker, M., Brown, P., Moore, D.: *Object-relational DBMSs - The Next Wave*, Second Edition, Morgan Kaufmann 1998.
- [6] SQL99: *ANSI/ISO/IEC 9075-1-1999 Database Languages SQL Part 1 Framework*.
- [7] SQL99: *ANSI/ISO/IEC 9075-2-1999 Database Languages SQL Part 2 Foundation*.
- [8] Webby, R., Seaman, C., Mendonca, M., Basili, V.R., Kim, Y.-M.: *Implementing an Internet-Enabled Software Experience Factory: Work in Progress*, ICSE'99 Workshop on Software Engineering over the Internet, Mai, 1999.
- [9] Westfechtel, B.: *Models and Tools for Managing Development Processes*, LNCS 1646, Springer, Berlin, 1999.

3.4 Eigene Vorarbeiten

Viele der in der vergangenen Förderungsperiode gewonnenen Erkenntnisse werden zur Erlangung der in der kommenden Periode angestrebten Ziele hilfreich sein. Nachfolgend sind einige wichtige Veröffentlichungen zu unseren Vorarbeiten aufgelistet. Für eine ausführlichere Darstellung unserer Arbeiten verweisen wir an dieser Stelle auf den entsprechenden Ergebnisbericht.

Da wir seit Anfang 1999 (weltweit der einzige universitäre) Partner von Informix in dessen Beta-Test-Programm für den *Informix Dynamic Server* sind, haben wir uns Kompetenz und praktisches Know-how für Einsatz, Erweiterung und Evaluation von ORDBVS erworben, die der Projektarbeit zugute kommen. Außerdem besitzen wir auf diesem Weg jeweils die neueste OR-Technologie für unsere Forschungs- und Entwicklungsarbeiten.

Literatur

- [10] Feldmann, R.L., Geppert, B., Mahnke, W., Ritter, N., Rößler, F.: *An ORDBMS-based Reuse Repository Supporting the Quality Improvement Paradigm - Exemplified by the SDL-Pattern Approach*, angenommen für TOOLS USA 2000, 34th International Conference & Exhibition, Santa Barbara, CA, July, 30 - August, 3, 2000.
- [11] Härder, T., Mahnke, W., Ritter, N., Steiert, H.-P.: *Generating Repository Managers for Cooperative Design Applications*, in: Proc. 2nd Int. Symposium on Cooperative Database Systems for Advanced Applications (CODAS'99), Keynote-Paper, Springer, Singapore, 1999, 177-191.
- [12] Härder, T., Mahnke, W., Ritter, N., Steiert, H.-P.: *Generating Versioning Facilities for a Design Data Repository Supporting Cooperative Applications*, Int. Journal on Intelligent & Cooperative Information Systems 9:1-2, 2000, 117-146.

- [13] Härder, T., Nink, U., Ritter, N.: *Generierte DB-Aufrufsschnittstellen - Anwendungsspezifische Zugriffsoptimierung durch Bindungsflexibilität*, Informatik - Forschung und Entwicklung 15:2, Springer-Verlag, 2000.
- [14] Käfer, W., Ritter, N., Schöning, H.: *Konfigurierungskonzepte für datenbank-basierte, technische Entwurfsanwendungen*, Informatik - Forschung und Entwicklung, 13:1, Springer-Verlag, 1998, 1-17.
- [15] Mahnke, W., Ritter, N., Steiert, H.-P.: *Towards Generating Object-Relational Software Engineering Repositories*, in: Proc. der GI-Fachtagung 'Datenbanksysteme in Büro, Technik und Wissenschaft' (BTW'99), Buchmann, A. (Hrsg.), Informatik aktuell, Springer-Verlag, Freiburg, März 1999, 251-270.
- [16] Mahnke, W., Steiert, H.-P.: *Zum Einsatzpotential von ORDBVS in Entwurfsumgebungen*, in: Proc. Fachtagung CAD'2000, Berlin, März 2000, 219-239.
- [17] Marder, U.: *Medienspezifische Datentypen für objekt-relationale DBMS: Abstraktionen und Konzepte*, in: Proc. der GI-Fachtagung 'Datenbanksysteme in Büro, Technik und Wissenschaft' (BTW'99), Buchmann, A. (Hrsg.), Informatik aktuell, Springer-Verlag, Freiburg, März 1999, 210-231.
- [18] Nink, U., Härder, T., Ritter, N.: *Generating Call-Level Interfaces for Advanced Database Application Programming*, in: Proc. 25th Int. Conf. on Very Large Data Bases (VLDB'99), Edinburgh, Sept. 1999, 575-586.
- [19] Ritter, N., Steiert, H.-P.: *Enforcing Modeling Guidelines in an ORDBMS-based UML Repository*, angenommen für International Resource Management Association Conference 2000 (Information Modeling Methods and Methodologies Track of IRMA 2000), Anchorage, Alaska, Mai 2000.
- [20] Ritter, N., Steiert, H.-P., Mahnke, W., Feldmann, R.: *An Object-Relational SE-Repository with Generated Services*, in: Proc. International Resource Management Association Conference 1999, Managing Information Technology Resources in Organizations in the Next Millenium (Computer-Aided Software Engineering Track of IRMA'99), IDEA Group Publ., Mai 1999, 986-990.
- [21] Steiert, H.-P.: *Towards a Component-based n-tier C/S-Architecture*, in: Proc. 3rd Int. Software Architecture Workshop (ISA W3), Orlando, Fl., ACM Press, 1998, 37-140.
- [22] Zhang, N., Härder, T.: *On a Buzzword "Extensibility" - What we have learned from the ORIENT Project*, in: Proc. Int. Database Engineering and Applications Symposium (IDEAS'99), Montreal, Canada, 1999, 360-369.
- [23] Zhang, W.P., Ritter, N.: *Measuring the Contributions of (O)RDBMS to Object-Oriented Software Development*, angenommen für Int. Database Engineering and Applications Symposium (IDEAS 2000), Yokohama, Japan, September 2000.

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

In diesem Abschnitt werden die konkreten Ziele, die A3 in der Förderungsperiode 2001-2003 verfolgt, genannt und die zu ihrer Erreichung geplanten Vorgehensweisen beschrieben.

Modellintegration, Erweiterbarkeit, Tiefenintegration

Die nahtlose und durchgängige Integration objektorientierter Konzepte in das relationale Datenmodell sowie geeignete Konzepte und Schnittstellen zur Erweiterbarkeit, die dem Anwender eine möglichst effiziente Integration von Anwendungsfunktionalität gestatten, beziehen sich auf eine eher grundlegende Gestaltung objekt-relationaler DBVS. Die Berücksichtigung der Anforderungen der SE-Domäne bei dieser Gestaltung ist das Langzeitziel von A3. Alle weiteren genannten Ziele und die zu ihrer Erreichung notwendigen Arbeiten tragen auch zur Erreichung dieses (übergeordneten) Zieles bei.

ORDB-gestütztes, integriertes SE-Repository

Der in der Förderungsperiode 1998-2000 entwickelte SFB-501-EDB-Prototyp soll in der kommenden Förderungsperiode sowohl in Teilprojekt-übergreifenden (Teams) als auch in Teilprojekt-spezifischen Experimenten eingesetzt und validiert werden. Dabei sollen insbesondere Erkenntnisse über die Eignung der definierten Datenstrukturen zur Verwaltung von Erfahrungen und der Web-basierten Benutzerschnittstelle gewonnen werden.

Diese Erkenntnisse sollen in die Neugestaltung des Ansatzes einfließen, die das Ziel der Bereitstellung eines integrierten, umfassende SE-Repository verfolgt. Dies bedeutet, dass nun der Schwerpunkt eindeutig auf die Integration von Erfahrungsdaten- und Produktdatenverwaltung sowie der Ablaufsteuerung gelegt wird. Dazu sind auf jeden Fall folgende Integrationsziele zu erreichen:

- Experiment-spezifische Produktdatenstrukturen müssen dynamisch in das Repository eingebracht werden können.
- Heterogene Entwurfswerkzeuge müssen an das Repository angeschlossen werden können.
- Sowohl Erfahrungs- als auch Produktdaten müssen (möglicherweise nach experiment-spezifischen Mechanismen) versioniert und konfiguriert werden können.
- Die Ablaufsteuerungskomponente MILOS ist mit der Erfahrungsdaten- und Produktdatenverwaltung zu einem umfassenden SE-Repository zu verschmelzen.

Hinsichtlich der Erreichung dieser (Integrations-)Ziele soll das bereits bestehende VirtualMedia-Framework im Rahmen des SE-Repository realisiert und evaluiert werden. Dieses erlaubt eine von dem eigentlichen Speicherungsformat weitgehend unabhängige und transparente Bereitstellung von Software-Artefakten in der vom anfragenden Benutzer (oder Entwurfswerkzeug) benötigten Repräsentation. Weiterhin soll die Nutzung von XML als Speicherungs- und Austauschformat für Software-Artefakte untersucht werden. Diese erscheint nützlich zu sein, um die zu erwartende Heterogenität der zu koppelnden Werkzeuge zu bewältigen.

Generische Methoden zur Entwicklung von ORDB-Anwendungen

Bei der in der Förderungsperiode 1998-2000 durchgeführten Gestaltung des SERUM-Frameworks zur Generierung anwendungsspezifischer Versionierungsfunktionalität hat es sich gezeigt, dass es sinnvoll ist, anwendungsspezifische Datenverwaltungsfunktionalität ausgehend von abstrakten Benutzerspezifikationen mit generischen Methoden zu entwickeln. Gegenstand der Generierung sind dabei sowohl Erweiterungen des ORDBVS als auch Application-Server-Funktionen (Mehr-Schichten-Architekturen) und Schnittstellenfunktionen. Dieser (Versionierungs-)Ansatz soll nun zu einem allgemeinen Ansatz konzipiert und prototypisch realisiert werden, der die Möglichkeit der Spezifikation anwendungsspezifischer Datenverwaltungsanforderungen in UML vorsieht und einen Generator zur Erzeugung der oben genannten Funktionalität erlaubt. Aus Aufwandsgründen kann eine experimentelle Validierung dieses Ansatzes erst in der nachfolgenden Förderungsperiode in Angriff genommen werden.

Flexibilisierung des OR-Verarbeitungskonzepts

Die mittels Modellintegration und Erweiterbarkeit durch objekt-relationale Systeme erreichte Flexibilität hat bisher kaum Einzug in die Gestaltung des Verarbeitungsmodells dieser Systeme gefunden. Dieses Defizit soll durch die Gestaltung eines flexiblen Verarbeitungskonzepts behoben werden. Die gegenwärtige Serverzentrierung soll jedoch nicht nur um die Möglichkeit der Client-basierten Verarbeitung von DB-Daten erweitert werden, sondern es soll dynamisch und automatisch entschieden werden können, ob eine zur Ausführung anstehende Anwendungsfunktion effizienter auf dem DB-Server oder dem DB-Client ausgeführt werden kann. Ein erstes Konzept liegt bereits vor; dieses soll nun prototypisch implementiert, experimentell validiert und konzeptionell verfeinert werden.

OR-Benchmark

Durch erste Messungen wurde bereits ein Vergleich von relationalen und objekt-relationalen DBVS hinsichtlich ihres Leistungsverhaltens, insbesondere in der Verwaltung persistenter Objekte, durchgeführt. Ausgehend von diesen Ergebnissen wird ein neuer Benchmark-Ansatz angestrebt. Dieser Ansatz soll es erlauben, nicht nur das Leistungsverhalten der DBVS selbst, sondern auch ihre Beiträge zum Leistungsverhalten von DB-basierten Anwendungssystemen und auch zur (in aller Regel objektorientierten) Entwicklung solcher Systeme zu beurteilen und zu vergleichen. Dazu genügt es nicht, so zu verfahren, wie es in gegenwärtig existierenden Benchmark-Ansätzen getan wird. Hier werden in aller Regel lediglich ein repräsentatives DB-Schema und eine Menge repräsentativer Anfragen vorgegeben. Durch Messungen wird dann das Leistungsverhalten von DBVS während der Abarbeitung der Anfragen erfasst. Ein neuer Ansatz sollte jedoch berücksichtigen, dass die Anforderungen, die komplexe Softwaresysteme an die Datenverwaltungskomponente stellen, im allgemeinen nicht direkt auf die DB-Schnittstelle abbilden lassen, sondern eine Abbildungsschicht erfordern. Ein Benchmark sollte deshalb so konzipiert sein, dass er auch den Aufwand für die Erstellung dieser Abbildungsschicht und den Aufwand, der wäh-

rend der Verarbeitung durch diese Abbildungsschicht verursacht wird, mit berücksichtigen kann. Hier sind durchaus Möglichkeiten der generischen Entwicklung von sowohl Abbildungsfunktionalität als auch Schnittstellenfunktionen für objektorientierte Programmiersprachen, wie sie auch in diesem Teilprojekt angestrebt werden, in die Konzeption des Benchmarks einzubeziehen.

Ein solcher neuer Benchmark-Ansatz soll ausgehend von dem oben genannten, bestehenden Ansatz konzipiert werden, ein zugehöriges Benchmark-System implementiert werden und erste Messungen an unterschiedlichen ORDBVS sollen durchgeführt werden.

Alle in diesem Abschnitt genannten Ziele sollen erreicht werden, in dem zunächst entsprechende Konzepte erarbeitet werden und anschließend die Machbarkeit durch Bereitstellung von Prototypen nachgewiesen wird.

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: (OR)DB-gestütztes, integriertes SE-Repository

In der Förderungsperiode 1998-2000 wurde in Zusammenarbeit mit Teilprojekt A1 (SE-Labor) eine (OR)DB-gestützte Erfahrungsdatenbank (SFB-501-EDB) entwickelt. Es handelt sich hierbei um einen ersten Prototyp, der direkt die Möglichkeiten eines kommerziell verfügbaren ORDBVS (hier IDS/UDO von Informix), das von uns als für diese Zwecke am geeignetsten erachtet wurde, ausnutzt. Mit dem Ende der Förderungsperiode 1998-2000 soll die Web-basierte Schnittstelle der SFB-501-EDB allgemein zugänglich gemacht werden und somit die Möglichkeit geschaffen werden, dass an Experimenten beteiligte SFB-Mitglieder Erfahrungsdaten in der SFB-501-EDB ablegen bzw. in der SFB-501-EDB gespeicherte Erfahrungsdaten einfach und effizient nutzen können. Ziel des EDB-Projektes über die gegenwärtige Förderungsperiode hinaus ist die umfassende Unterstützung des nach dem *Quality-Improvement-Paradigm* (QIP) ablaufenden Softwareentwicklungsprozesses. Dies beinhaltet die integrierte Verwaltung von Erfahrungsdaten und Produktdaten, die Bereitstellung von Schnittstellen, die alle QIP-Phasen geeignet unterstützen sowie die Interoperabilität der im SE-Labor bereitstehenden Werkzeuge mit der SFB-501-EDB.

Arbeitspaket A1: Application-Server-Architektur

Der gegenwärtige EDB-Prototyp wurde unter Ausnutzung des Web-Data-Blades von Informix realisiert, das eine dynamische Einbettung von DB-Daten in HTML-Seiten unterstützt. Da sich jedoch einige Dienste, die zur Unterstützung der SE-Anforderungen notwendig sind, nicht ausschließlich durch Erweiterung des DBS realisieren lassen, sondern eine Zusatzebene oberhalb der DB-Schnittstelle erfordern (z. B. Versionierung), soll der Architekturansatz um eine Application-Server-Ebene oberhalb des DBS erweitert werden. Application-Server erlauben nicht nur eine bessere Abbildbarkeit höherer Dienste (wie Versionierung) auf die DB-Schnittstelle, sondern tragen i. allg. auch zu besserem Leistungsverhalten bei. Es ist zu erwarten, dass auch

die meisten der in den nachfolgenden Arbeitspaketen verfolgten Ziele eine Realisierung im Rahmen einer Application-Server-Architektur erfordern.

Personalbedarf: 22 PM

Arbeitspaket A2: Ähnlichkeitsbasierte Suche in ORDBVS

Der gegenwärtig verfügbare Prototyp der SFB-501-EDB hat gezeigt, dass die Erweiterungsmöglichkeiten des verwendeten objekt-relationalen DBVS die Realisierung ähnlichkeitsbasierter Suche unterstützen. Dazu wurden benutzerdefinierte Funktionen für die ähnlichkeitsbasierte Suche implementiert, die direkt im DB-System ausgeführt werden können. Über eine Parametrisierung dieser Funktionen können verschiedene Ähnlichkeitsmaße nach der Spezifikation der Benutzer ausgewertet werden. Damit konnte nachgewiesen werden, dass diese Anforderung von ORDBVS besser als von anderen Arten von DBVS, bei denen die Abwicklung der Suche erst im Client erfolgt, erfüllt werden kann.

Werkzeuge, die auf Case-Based Reasoning beruhen (CBR-Tools), lassen sich ebenfalls zur ähnlichkeitsbasieren Suche einsetzen, wobei die zu durchsuchenden Daten nicht in einem DB-System verwaltet, sondern zur Suche im Hauptspeicher gehalten werden. Diese Werkzeuge wurden speziell zum Zweck einer unscharfen Suche entwickelt, wobei Aspekte der Datensicherung, Transaktionsverarbeitung usw. vernachlässigt wurden. Während diese Probleme durch die Verwendung eines ORDBVS im Falle der SFB-501-EDB gelöst werden, stellt sich die Frage, inwieweit die SFB-501-EDB einem Vergleich mit diesen spezialisierten Werkzeugen im Bereich der ähnlichkeitsbasierten Suche standhält. Dazu ist zum einen zu untersuchen, welche Möglichkeiten der Suche bei den CBR-Tools geboten werden und inwieweit diese von der SFB-501-EDB bereits unterstützt werden bzw. unterstützt werden können. Zum anderen ist ein Leistungsvergleich zwischen der ähnlichkeitsbasierten Suche der SFB-501-EDB mit der Suche kommerziell erwerblicher CBR-Tools durchzuführen.

Personalbedarf: 14 PM

Arbeitspaket A3: Transformationsunabhängigkeit durch VirtualMedia

Im *VirtualMedia*-Projekt wird eine Integration von ORDBVS und Medien-Servern angestrebt, welche die Verwaltung großer, (aus DBVS-Sicht) unstrukturierter Dokumente sehr viel flexibler und mächtiger macht, als dies bei heutigen DBVS der Fall ist. Die vom VirtualMedia Framework unterstützten Eigenschaften können zusammengefasst als *Transformationsunabhängigkeit* bezeichnet werden, da sie es Anwendungen ermöglichen, von anderen Anwendungen generierte Dokumente zu öffnen und zu bearbeiten, ohne deren Speicherungsformat zu kennen.

Die durch VirtualMedia angesprochenen Probleme treten häufig in der Erfahrungsdatenbank auf, denn in der SFB-501-EDB sollen Dokumente, die von externen SE-Werkzeugen erzeugt und bearbeitet werden, (auch) in ihrem proprietären Format verwaltet werden. Um diese Daten auch mit einfacheren Werkzeugen, z. B. Browsern, auswerten zu können, ist es i. allg. notwendig, verschiedene physische Repräsentationen dieser Daten zu speichern. Die durch VirtualMedia realisierte Transformationsun-

abhängigkeit kann dieses Problem jedoch wesentlich eleganter lösen, indem die Dokumente je nach Bedarf der anfragenden Anwendung in der jeweils optimalen externen Repräsentation ausgeliefert werden, ohne dass deshalb die Notwendigkeit besteht, alle möglichen Repräsentationen physisch vorzuhalten.

Personalbedarf: 18 PM

Arbeitspaket A4: Integrierte Produktdatenverwaltung

Zweifellos eignen sich die Fähigkeiten von ORDBVS nicht nur zur Verwaltung von Erfahrungsdaten, sondern auch zur Verwaltung von Produktdaten. Darüber hinaus gibt es eine ganze Reihe von guten Gründen für eine integrierte Verwaltung dieser verschiedenen Arten von Daten in einem einzigen System (das natürlich trotzdem ein verteiltes System sein kann). Beispielsweise gehen Entwicklungsprozesse häufig von bereits vorhandenen, in der Erfahrungsdatenbank gehaltenen Artefakten aus, die im Laufe des Prozesses verfeinert/angepasst/verändert werden. In ähnlicher Weise werden sehr häufig (Zwischen-)Ergebnisse von Entwicklungsprozessen als Erfahrungen freigegeben. Darüber hinaus bestehen zwischen den Produktdaten eines laufenden Entwicklungsprozesses und den Erfahrungsdaten oft Abhängigkeiten, die bei integrierter Verwaltung als semantische Beziehungen systemseitig gewartet werden könnten. Daher soll in diesem Arbeitspaket die bereits in Zusammenarbeit mit Teilprojekt D1 erarbeitete Möglichkeit der Integration einer Produktdaten- und Erfahrungsdatenverwaltung zu einem allgemeinen Konzept ausgebaut werden. Deshalb ist die Erfahrungsdatenbank um entsprechende Funktionalität zu erweitern.

Personalbedarf: 10 PM

Arbeitspaket A5: Integration mit MILOS

Die bereits bestehenden Konzepte zur Integration von im SE-Labor bereitstehenden Werkzeugen mit der SFB-501-EDB sollen hinsichtlich des neuen Architekturansatzes überarbeitet werden, um die Integration dann in Kooperation mit den betroffenen Teilprojekten durchzuführen. Insbesondere die Integration der durch die SFB-501-EDB unterstützten Datenverwaltung mit der durch MILOS unterstützten Prozesskontrolle wird in Zusammenarbeit mit den Projekten A1, A2, B2 verfolgt. Der in der Förderungsperiode 1998-2000 ausgearbeitete Vorschlag einer Integration soll in der kommenden Förderungsperiode verfeinert, realisiert und evaluiert werden.

Da es sich bei MILOS um eine datenintegrierte Ablaufsteuerung handelt, d. h., MILOS zur Zeit eine eigene Kontrolle der Applikationsdaten (Produktdaten) vornimmt, muss dieses Arbeitspaket in enger Abstimmung mit dem vorgenannten ablaufen. Nach einer entsprechenden Integration können nicht nur Prozessmodelle und Projektpläne als Erfahrungen verwaltet werden, sondern es können auch Prozesszustände beschreibende Kontrolldaten, die von MILOS benötigt werden, gehalten und mit entsprechenden Produktdaten(zuständen) assoziiert werden. Letzteres ist insbesondere einer empirischen Bewertung von Experimenten dienlich.

Personalbedarf: 10 PM

Arbeitspaket A6: Nutzung von XML

Wie bereits angesprochen, ist die SFB-501-EDB mit einer Web-Schnittstelle ausgestattet. Die gegenwärtige Entwicklung der Web- wie auch der DB-Technologie beschäftigt sich sehr stark mit der Nutzung von XML sowohl zum Austausch als auch zur Verwaltung semi-strukturierter Daten. Daher soll untersucht werden, inwieweit sich XML zur Speicherung und zum Austausch von Erfahrungsdaten eignet und inwieweit Nutzen aus der XML-Unterstützung der kommerziell verfügbaren ORDBVS gezogen werden kann. Darüber hinaus ist zu prüfen, inwieweit die Nutzung von XML die vielfältigen, in diesem Arbeitsschwerpunkt vorgesehenen Integrationsbemühungen erleichtern könnte.

Personalbedarf: 12 PM

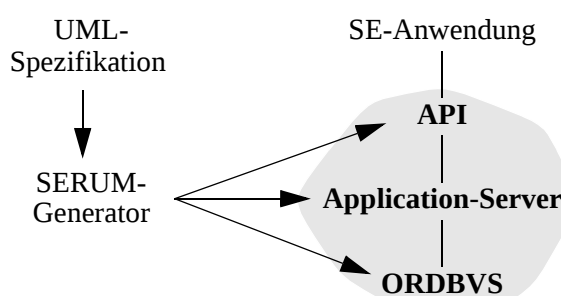
Arbeitsschwerpunkt B: Generische Entwicklung von (OR)DB-Anwendungen

In der Förderungsperiode 1998-2000 hat sich gezeigt, dass die vielfältigen Anforderungen, die in modernen Softwaresystemen an die Datenverwaltung gestellt werden, am flexibelsten durch ein Framework erfüllt werden können, das die gezielte Anpassung der Datenverwaltungsfunktionalität bzw. die anwendungsspezifische *Generierung* von Datenverwaltungsfunktionalität unterstützt. Ein solches Framework wurde bereits unter dem Namen SERUM konzeptualisiert und insbesondere im Hinblick auf die generische Bereitstellung anwendungsspezifischer Versionierungsfunktionalität ausgestaltet. Von der weiteren Ausgestaltung und prototypischen Realisierung von SERUM erwarten wir insbesondere Erkenntnisse darüber, inwieweit die im SFB betrachteten generischen Methoden der Softwareentwicklung im Bereich der Entwicklung von (OR)DB-Anwendungen (z. B. Generierung von ORDBVS-Erweiterungen) eingesetzt werden können.

Arbeitspaket B1: Generierung von Datenverwaltungsfunktionalität ausgehend von UML-Spezifikationen

Als Eingabe für den Generierungsprozess sollen UML-Spezifikationen dienen, da UML das Paradigma der Objektorientierung geeignet umsetzt und sich somit für die Beschreibung anwendungsspezifischer Aspekte besonders eignet. Hinsichtlich des Ergebnisses des Generierungsprozesses hat sich herausgestellt, dass aus

Gründen der Effizienz und der sinnvollen Nutzung der Erweiterungsmechanismen von ORDBVS eine Application-Server-Architektur anzustreben ist, wie sie in nebenstehender Abbildung (grau hinterlegt) angedeutet ist. Daher sind die bestehenden Konzepte durch Beantwortung der Frage, nach welchen Prinzipien Elemente von UML-Spezifikationen auf ORDBVS-Erweiterungen (z. B. benutzerdefinierte Funktionen/Datentypen), Application-Server-Module bzw. für die Anwendung zugängliche



Schnittstellen abgebildet werden, zu verfeinern. Dies beinhaltet auch die Entscheidung, welche Teile der benötigten Anwendungsfunktionalität sinnvollerweise als Erweiterungen des ORDBVS realisiert werden sollten und welche darüber anzusiedeln sind.

Personalbedarf: 26 PM

Arbeitspaket B2: Nutzung von Frameworks

Während sich der vorgenannte Punkt mit grundlegenden Mechanismen der Generierung von Datenverwaltungsfunktionalität (z. B. Versionierungsdienste) beschäftigt, soll hier der Schwerpunkt auf einer Unterstützung der Spezifikationsphase liegen. SERUM-Frameworks beinhalten unter anderem *Design Patterns*, die eine automatische Verfeinerung und Anreicherung von initialen, vom Benutzer gegebenen UML-Spezifikationen zu solchen Spezifikationen erlauben, die auch anwendungsunabhängige Aspekte umfassen und gültige Eingaben für den SERUM-Generator darstellen. Beispielsweise kann man es auf diese Weise dem Benutzer erlauben, ein Produktdatenmodell vollkommen unabhängig von Versionierungsaspekten zu spezifizieren und die Versionierungsinfrastruktur weitgehend automatisch durch Anwendung von *Design Patterns* zu ergänzen.

Personalbedarf: 15 PM

Arbeitspaket B3: Generierung von anwendungsspezifischen Aufrufschnittstellen

Es wurde bereits ein Konzept vorgestellt, das es erlaubt, Eigenschaften strukturell objektorientierter Datenmodelle und deren Anfragesprachen sowie Anwendungswissen auszunutzen, um speziell zugeschnittene Anwendungsprogrammierschnittstellen für die objektorientierte Anwendungsentwicklung zu generieren. Damit kann eine anwendungsspezifische Optimierung erreicht werden. Diese Konzepte sollen auf das objekt-relationale Datenmodell übertragen werden. Somit lässt sich die Möglichkeit schaffen, auf spezielle Anwendungen, z. B. objektorientierte Softwareentwicklung, zugeschnittene Schnittstellenfunktionen für die Datenverwaltung zu generieren. Insbesondere durch die geeignete Nutzung früher und später Bindungen von Schnittstellenfunktionen an DB-Typen und -Verarbeitungsmechanismen kann ein großes Optimierungspotential geboten werden.

Personalbedarf: 15 PM

Arbeitspaket B4: Flexibles Verarbeitungskonzept

In der gegenwärtigen Förderungsperiode wurde ein Konzept erarbeitet, das es erlaubt, dynamisch zu entscheiden, ob eine zur Ausführung anstehende, DB-Daten manipulierende Anwendungsfunktion auf dem DB-Server oder dem Client-Cache ausgeführt werden soll. Dazu wird das System in die Lage versetzt, Folgen und Aufwand von Funktionsausführungen (Datentransfer, Ausführungszeit) zu protokollieren, um spätere Funktionsfolgen zu optimieren. Mit Hilfe der Protokolle kann dann dynamisch mit heuristischen Methoden entschieden werden, ob eine Funktion auf Client oder Server abgewickelt wird. Dieses Konzept soll in der kommenden Periode verfeinert,

prototypisch implementiert und durch empirische Untersuchungen auf seine Effektivität untersucht werden.

Personalbedarf: 15 PM

Arbeitsschwerpunkt C: Begleitende Untersuchungen

Wie bereits angeklungen, soll in diesem dritten Arbeitsschwerpunkt das Potential objekt-relationaler DB-Technologie hinsichtlich der Unterstützung von SE-Prozessen untersucht werden. Darüber hinaus soll dazu beigetragen werden, dass (künftige) ORDBVS die Anforderungen der SE-Domäne angemessen berücksichtigen.

Arbeitspaket C1: Erweiterbarkeit

Obwohl mit dem Standard SQL:1999 eine Standardisierung des objekt-relationalen Datenmodells und der zugehörigen Sprache erreicht worden ist, ist die Frage nach der sinnvollen Integration von objektorientierten mit relationalen Konzepten noch nicht endgültig geklärt. Auch die optimale Umsetzung der Sprache durch geeignete (interne) DBVS-Module ist noch unklar. Hinsichtlich der Erweiterbarkeitseigenschaft ist insbesondere zu klären, welche Schnittstellen dem Benutzer zur Erweiterung des DBVS angeboten werden sollten und wie die anwendungsspezifischen Erweiterungs-module effektiv in die interne Datenverarbeitung integriert werden können. Hierzu sollen die im ORIENT-Projekt, das noch in der gegenwärtigen Förderungsperiode abgeschlossen wird, gewonnenen Erkenntnisse ausgenutzt werden.

Personalbedarf: 10 PM

Arbeitspaket C2: Effizienzuntersuchungen

Nachdem in der gegenwärtigen Förderungsperiode ein erstes Konzept für einen Benchmark zur Bewertung des Leistungsverhaltens von ORDBVS hinsichtlich der Unterstützung von (objektorientierten) Entwicklungsanwendungen erarbeitet wurde, soll dieses in der kommenden Periode verfeinert, vollständig implementiert und zur Bewertung verschiedener ORDBVS herangezogen werden. Bei der angesprochenen Verfeinerung geht es im Vergleich zu existierenden Benchmark-Ansätzen zunächst darum, die wirklichen Datenverwaltungsanforderungen von Softwaresystemen, die objektorientiert entwickelt werden (z. B. Anforderungen an ein persistentes Objektsystem), überhaupt zu erkennen. Ausgehend von diesen Anforderungen sind dann Regeln zu entwickeln, wie für ein gegebenes DBVS die geforderte Datenverwaltungsfunktionalität realisiert werden sollte. Letzteres kann durch direkte Nutzung der Möglichkeiten des DBVS, durch Erweiterung des DBVS oder durch Realisierung einer Abbildungsschicht (Application Server) oberhalb der DBVS-Schnittstelle geschehen. Die jeweils effizienteste Lösung ist auszuwählen und durch Simulation eines repräsentativen Anwendungsszenarios quantitativ zu beurteilen.

Personalbedarf: 18 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausstattung						
N. Ritter	A, B, C					
H.-P. Steiert	B1				B2	
		B3				
	B4					
N. Zhang	C1					
Ergänzungsausstattung						
W. Mahnke	B1				B2	
	A1, A4, A5					
	A2					
U. Marder	A1, A3, A4, A5					
W. Zhang, StHkr.	C2					
N.N., StHKr.	B					
N.N., StHKr.	A					

Im Zeitplan ist ein Aufwand von wenigstens 8 Std. / Woche für SFB-übergreifende Themenstellungen berücksichtigt, die jedes Teilprojekt zur Verfügung stellt. Diese Arbeiten werden in Teams organisiert und im Arbeitsbericht gesondert beschrieben.

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Gemäß der Zuordnung dieses Teilprojekts zum SFB-Projektbereich A wird der realisierte Prototyp der Erfahrungsdatenbank in enger Zusammenarbeit mit Teilprojekt A1 im Software-Labor zur Verfügung gestellt und dort gewartet. Um eine geeignete Evaluierung dieses Ansatzes zu erreichen, wird die Nutzung dieses Prototyps sowohl in Teilprojekt-übergreifenden als auch in Teilprojekt-spezifischen Experimenten unterstützt. Wie oben beschrieben, beziehen sich die weiteren Arbeiten des Arbeitsschwerpunktes A auf einen Ausbau der Erfahrungsdatenbank zu einem umfassenden SE-Repository. Die dazu notwendigen Integrationsarbeiten, insbesondere die Bestrebungen der Produktdatenintegration und der Werkzeuganbindung, bedürfen der engen Zusammenarbeit mit fast allen Teilprojekten des SFB. Besonders zu betonen ist jedoch die weitere Zusammenarbeit mit den Teilprojekten A1, A2, B2 und D1, die das Ziel der Integration der Produktdaten- und Erfahrungsdatenverwaltung sowie der Ablaufsteuerung verfolgen. Weiterhin ist hier eine geplante Kooperation mit dem Teilprojekt B10 zu nennen. B10 strebt eine Verallgemeinerung der bisher auf den Bereich der Betriebssystemkomponenten zugeschnittenen Entwicklungsmethodik an. In Kooperation der Teilprojekte A1, A3 und B10 ist zu prüfen, ob die SFB-501-EDB

diese verallgemeinerte Methodik unterstützt oder entsprechende Anpassungen erforderlich werden.

Arbeitsschwerpunkt B strebt Methodiken der Entwicklung von anwendungsspezifischer Datenverwaltungsfunktionalität mit generischen Methoden an. Offenbar besteht damit ein Zusammenhang mit den im SFB-Projektbereich B entwickelten Methodiken. Hierzu ist zu sagen, dass die in A3 angestrebten Vorgehensweisen zur Unterstützung der in den B-Teilprojekten entwickelten Vorgehensweisen gedacht sind, soweit es um die generische Entwicklung von Datenverwaltungsfunktionalität geht.

Im Arbeitsschwerpunkt C werden grundsätzliche Aspekte der geeigneten Gestaltung objekt-relationaler DBVS bearbeitet. Wie bereits angesprochen, werden hiermit langfristige Ziele verfolgt, die eher mittelbar in die von A3 bereitzustellenden Konzepte und Systeme einfließen werden.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	2	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
	.		StdHkr	3	61.200	StdHkr.	3	61.200	StdHkr.	3	61.200
	zus.:	2	zus.:	5	270.000	zus.:	5	270.000	zus.:	5	270.000
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		3.000	(522)		3.000	(522)		3.000
			zusammen		3.000	zusammen		3.000	zusammen		3.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Härder, T.; Prof. Dr.	Prakt. Inf.	FB Inf., Uni KL	5	1.1.98	
	(2) Ritter, Dr.	Prakt. Inf.	Fb Inf., Uni KL	10	1.1.98	
	(3) H.-P. Steiert	Prakt. Inf.	FB Inf., Uni KL	20	1.1.98	
	(4) Zhang, N., M. Sc.	Prakt. Inf.	Fb Inf., Uni KL	10	1.1.98	
Ergänzungsausstattung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Mahnke, W., Dipl.-Inform.	Prakt. Inf.	FB Inf., Uni KL	38,5	15.3.98	Ila
	(2) Marder, U., Dipl.-Inform.	Prakt. Inf.	FB Inf., Uni KL	38,5	1.3.98	Ila
	(3) Zhang, W.P., StHkr.	Prakt. Inf.	FB Inf., Uni KL	19		
	(3) N.N., StHkr.	Prakt. Inf.	FB Inf., Uni KL	19		
	(3) N.N., StHkr.	Prakt. Inf.	FB Inf., Uni KL	19		
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Der erste wissenschaftliche Mitarbeiter (Steiert) ist für die Bearbeitung der Arbeitspakete des Arbeitsschwerpunktes B vorgesehen.

Der zweite wissenschaftliche Mitarbeiter (N. Zhang) ist für die Bearbeitung des Arbeitspakets C1 vorgesehen.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Der erste wissenschaftliche Mitarbeiter (Mahnke) ist für die Bearbeitung der Arbeitspakete A1, A2, A4 und A5 vorgesehen und unterstützt darüberhinaus den Arbeitsschwerpunkt B durch Bearbeitung der Arbeitspakete B1 und B2.

Der zweite wissenschaftliche Mitarbeiter (Marder) ist für die Bearbeitung der Arbeitspakete A1, A3, A4 und A5 vorgesehen.

Die erste Hilfskraft (W. Zhang) ist für die Durchführung der in Arbeitspaket C2 vorgesehenen Leistungsmessungen vorgesehen.

Die zweite Hilfskraft ist für Implementierungsarbeiten an der SFB-501-EB vorgesehen.

Die dritte Hilfskraft ist für Implementierungsarbeiten im Arbeitsschwerpunkt B vorgesehen.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausstattung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausstattung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausstattung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausstattung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausstattung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
	-	-	-
Summe	-	-	-

Projektbereich B

**Generische Modelle und
Methoden der Systementwicklung**

3 Projektbereich B: Generische Modelle und Methoden der Systementwicklung

3.01 Zusammenfassende Darstellung des Projektbereiches

In diesem Projektbereich werden generische Prozess- und Produktmodelle untersucht sowie Methoden für ihre werkzeuggestützte Erstellung und Wiederverwendung entwickelt.

In den Teilprojekten B1 und B2 entsteht ein generischer Prozessrahmen, der alle Software-Entwicklungsaktivitäten abdeckt. Dabei steht im Vordergrund die Verknüpfung und Integration von existierenden Entwicklungsmethoden und -werkzeugen mit dem Ziel, horizontale und vertikale Verfolgbarkeit von Entwurfsentscheidungen zu gewährleisten. Die Gewinnung, Strukturierung und Wiederverwendung von dynamisch anfallendem Erfahrungswissen ist ein besonderes Anliegen von B2, während in B1 der Aspekt der Qualitätssicherung durch entsprechende Qualitätsmodelle und -sicherungsprozesse einen besonderen Stellenwert besitzt.

In den Teilprojekten B4, B5, B10 und B12 werden generische Produktmodelle und dazu passende Teilprozesse für die Anwendungsfelder Kommunikationssysteme (B4), Betriebssysteme (B5) und Gebäudeautomationssysteme (B10 und B12) untersucht. Alle Anwendungsfelder stellen spezifische Anforderungen an generische Architektur- und Komponententechnologien und ergänzen sich deshalb sinnvoll. B10 und B12, aber auch teilweise D1, konkurrieren in Bezug auf geeignete Softwarearchitekturansätze für das Anwendungsfeld Gebäudeautomation miteinander. Ein wesentlicher Bestandteil im fortgeschrittenen Stadium dieser Teilprojekte wird deshalb auch eine vergleichende Bewertung der unterschiedlichen Ansätze sein. Die Erkenntnisse von C1 hinsichtlich einer möglichst weitgehenden Formalisierung von Architekturwissen werden in diesen Projekten nach Möglichkeit berücksichtigt.

Mit dem Teilprojekt B11 verstärkt der SFB seine Anstrengungen, nichtfunktionale Anforderungen an ein System am wichtigen Beispiel der Zuverlässigkeit auf systematische Weise in eine Systemstruktur umzusetzen. Der Schwerpunkt des Teilprojekts liegt dabei auf dem Prozess und nicht auf den Lösungskonzepten für Fehlertoleranz, die als gegeben angenommen werden.

Aus dem Projektbereich B können sich spezielle Fragestellungen an benötigte Beschreibungstechniken ergeben, bei deren Beantwortung das Teilprojekt C1 Hilfeleistung leistet. Werkzeug-Prototypen werden systematisch in den Prozessrahmen integriert und stehen damit im Teilprojekt A1 für Anwendungen zur Verfügung.

3.02 Bisherige und beantragte Förderung des Projektbereiches

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	2296,0	30,0	50,0	2376,0
1998	806,4	13,5	-	819,9
1999	825,6	13,5	-	839,1
2000	730,8	13,5	-	744,3
Zwischensumme	4658,8	70,5	50,0	4779,3
2001	1518,0	23,0	25,0	1566,0
2002	1518,0	21,0	-	1539,0
2003	1518,0	21,0	-	1539,0

(jeweils Ergänzungsausstattung, Beträge in DM 1000,-)

3.1 Allgemeine Angaben zum Teilprojekt B1

3.1.1 Thema

Generische Modellierung von Prozessen und Experimenten

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Software Engineering

3.1.3 Leiter¹

Prof. Dr. Dieter Rombach
geb. 6.6.1953
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2895
Fax: 0631/205-3331
rombach@informatik.uni-kl.de

Die Stelle des Leiters ist befristet ☒ ja ☐ nein ☐ ja, bis zum

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ ja	☒ nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ ja	☒ nein
Tierversuche	☐ ja	☒ nein
gentechnologische Untersuchungen	☐ ja	☒ nein

1. Die Aufnahme von Dipl.-Inform. Jürgen Münch als Projektleiter ist nach Abschluss seiner Dissertation vorgesehen.

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	351,2	3,0	0	354,2
1998	100,8	3,0	0	103,8
1999	103,2	3,0	0	106,2
2000	104,4	3,0	0	107,4
Zwischensumme	308,4	9,0	0	317,4
2001	249,6	3,0	0	252,6
2002	249,6	3,0	0	252,6
2003	249,6	3,0	0	252,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Software-Entwicklungsprozesse bestimmen die Qualität der resultierenden Software. Es gibt eine Vielzahl von Prozessen sowie Entwicklungsansätzen (Techniken, Methoden und Werkzeuge), die im Rahmen dieser Prozesse eingesetzt werden können. Bei der Projektplanung stellt sich somit folgende Aufgabe: Welcher Prozess und welche Entwicklungsansätze sind geeignet, vorgegebene Projektziele (z. B.: hohe Zuverlässigkeit) in einer bestimmten Anwendungsdomäne (z. B.: Gebäudeautomation) unter gegebenen Randbedingungen (z. B.: hohe Erfahrung der Entwickler) zu erreichen? Zur Lösung dieser Planungsaufgabe sind Erfahrungswerte bezüglich der Eignung aller in Frage kommender Entwicklungstechniken sowie -prozesse auf die relevanten Projektziele in dieser Domäne unter den gegebenen Randbedingungen notwendig. Experimente unterschiedlicher Form sind geeignet, derartige Erfahrungswerte zu erhalten.

Langfristiges Ziel dieses Teilprojekts ist die Entwicklung einer generisch ausgelegten Experimentiermethodik für die Gewinnung detaillierter Erfahrungswerte.

In der ersten Förderungsperiode wurde eine initiale, überwiegend manuelle experimentelle Methodik für die Durchführung von Experimenten und Fallstudien erstellt. Insbesondere wurden Techniken und Prototypwerkzeuge für die sichten-basierte Erstellung und explizite Repräsentation von Software-Entwicklungsprozessen (in Form von sog. Prozessmodellen) entwickelt. Des Weiteren wurden Techniken zur Definition von Messzielen, zur Ableitung von Metriken sowie zur Instrumentierung von Prozessen und Produkten entwickelt. Für die Ablage und Wiederverwendung der Ergebnisse von Experimenten wurde ein Schema bereitgestellt.

In der zweiten Förderungsperiode wurden diese Methoden, Techniken und Werkzeuge erprobt und optimiert. Die Optimierung erfolgte einerseits hinsichtlich der

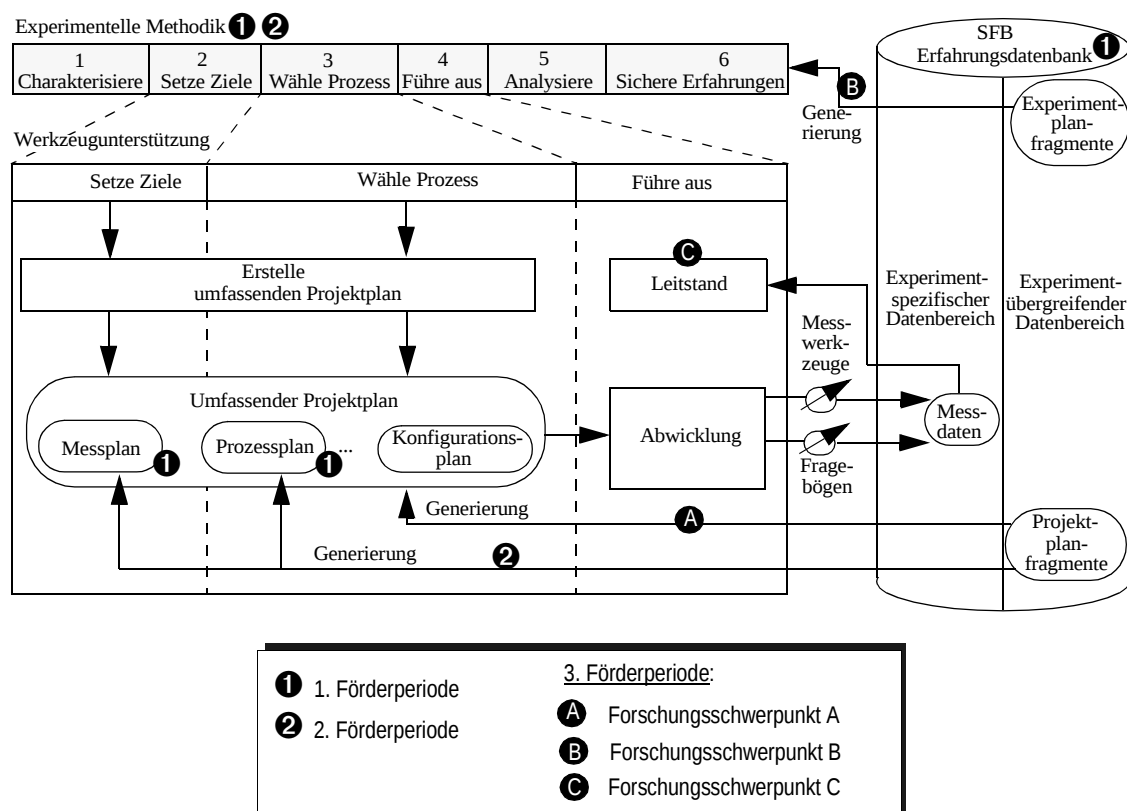


Abb. B1.1: Generisch ausgelegte Experimentiermethodik

Erhöhung der Wiederverwendbarkeit von Messplänen und Prozessmodellen durch deren generische Auslegung für das Anwendungsgebiet „Gebäudeautomation“. Andererseits wurde die Wiederverwendbarkeit von Ergebnissen vergangener Experimente durch deren Ablage im experimentübergreifenden Datenbereich der Erfahrungsdatenbank des SFB erhöht.

In der dritten Förderperiode wird die Generierung umfassender Projektpläne angestrebt, wobei verschiedene Sichten (z. B.: Messsicht, Prozesssicht) konsistent integriert werden können. Des Weiteren soll die Experimentiermethodik generisch ausgelegt werden, sodass eine flexible Anpassung an verschiedene Experimentierkontexte und -ziele erleichtert wird. Diese und bisherige Forschungsaufgaben des Teilprojekts B1 betreffen im wesentlichen Aspekte der *Projektplanung* und der *methodischen Fundierung von Experimenten*. In der dritten Förderperiode sollen zusätzlich Forschungsaufgaben bearbeitet werden, die im Bereich der *Abwicklung* von Projekten und Experimenten angesiedelt sind. Zur verbesserten Abwicklung von Projekten soll ein Leitstand zur messbasierten Projektsteuerung eingerichtet werden. Die Erweiterung des Forschungsumfangs des Teilprojekts B1 ergibt sich aus der Notwendigkeit, grundlegende Prinzipien der bisher entwickelten Ansätze systematisch und effektiv auf den Bereich der Abwicklung zu übertragen. Für diese Aufgaben wird eine zweite wissenschaftliche Mitarbeiterstelle benötigt.

Aufbauend auf den Ergebnissen der ersten beiden Förderungsperioden sollen die folgenden, sich ergänzenden Forschungsschwerpunkte bearbeitet werden:

- a) Entwicklung einer Methodik zur Generierung umfassender Projektpläne unter Einbeziehung expliziter Qualitätsmodelle;
- b) Entwicklung einer Methodik zur Generierung von Experimentfamilien;
- c) Untersuchung, Realisierung und Erprobung von Techniken zur automatisierten Datenerfassung, -interpretation, -präsentation und Qualitätssicherung im Rahmen eines Prozessleitstands (dies erlaubt eine On-Line-Unterstützung für das Projektmanagement und die Qualitätssicherung).

Im Falle einer Durchführung von Industrieprojekten in der dritten Förderungsperiode wird neben einer SFB-internen Evaluierung der Ergebnisse des Teilprojekts eine Extern-Erprobung in einem industriellen Anwendungsprojekt angestrebt.

3.3 Stand der Forschung

Die nachfolgende Veröffentlichungsliste gibt einen kurzen, repräsentativen Überblick über den derzeitigen Stand der Forschung in den für das Teilprojekt B1 relevanten Themengebieten. Eine aktuelle Übersicht über derzeit existierende Ansätze zur *Generierung umfassender Projektpläne* enthält [5]. Neben den hier dargestellten Ansätzen gewinnen bei der Wiederverwendung von Prozesswissen in jüngster Zeit Prozessmuster-basierte Ansätze eine zunehmende Bedeutung [1][4]. Eine Übersicht aktueller Forschungsansätze speziell für Prozessmodellierungsansätze enthält [3]. Wesentliche Forschungsarbeiten zum Themengebiet *Generierung von Experimentfamilien* sind in [2] beschrieben. Hier finden sich einerseits existierende Experimentfamilien für ausgewählte Software-Entwicklungstechniken, andererseits wird die Experimentierhistorie beschrieben, die zur Erstellung dieser Familien führte. Eine aktuelle Übersicht derzeit existierender Experimentiertechniken gibt [8]. Zum Themengebiet *Prozessleitstand* gibt es eine Vielzahl von Monitoring-, Controlling- und Datenpräsentationstechniken. Diese Ansätze sind jedoch maßgeschneidert für Domänen (z. B.: Produktionsprozesse, Geschäftsprozesse), die sich in wesentlichen Charakteristika (z. B.: Kreativität der Prozesse, Datenumfang) von der Software-Entwicklung unterscheiden. Ein rudimentär Ansatz aus der Domäne "Software-Entwicklung" ist der SME-Ansatz des NASA-SEL, der im WebMe-Ansatz weiterentwickelt wurde [6][7].

Literatur

- [1] Scott W. Amber. *Process Pattern: Building Large-Scale Systems Using Object Technology*. Cambridge University Press, 1998.
- [2] V. Basili, F. Shull und F. Lanubile: *Building Knowledge through Families of Experiments*, IEEE Transactions on Software Engineering, 1999.

- [3] Gianpaolo Cugola und Carlo Ghezzi: *Software Processes: a Retrospective and a Path to the Future*. Software Process - Improvement and Practice, 4(3): 101-123, September 1998.
- [4] Hajimu Iida: *Pattern-oriented Approach to Software Process Evolution*. In Proceedings of the International Workshop on Process of Software Evolution (IWPSE'99), Fukuoka City, Japan, 16.-17. Juli 1999.
- [5] Jürgen Münch: *Anpassung von Vorgehensmodellen im Rahmen ingenieurmäßiger Softwarequalitätssicherung*. In Tagungsband des 6. Workshops der Fachgruppe 5.1.1 (GI): Vorgehensmodelle, Prozessverbesserung und Qualitätsmanagement, Kaiserslautern, 19.-20. April, 1999.
- [6] R. Tesoriero und M. V. Zelkowitz: *WebME: A Web-based tool for Data Analysis and Presentation*, IEEE Internet Computing, September/Oktober 1998.
- [7] R. Tesoriero und M. V. Zelkowitz: *Using the Internet to Combine and Analyze Distributed Software Engineering Data*. Software Engineering over the Internet Workshop, April 1998.
- [8] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslen: *Experimentation in Software Engineering. An introduction*. The Kluwer International Series in Software Engineering, Kluwer Academic Publishers, Boston 2000.

3.4 Eigene Vorarbeiten

Die in der kommenden Förderungsperiode geplanten Arbeiten bauen größtenteils auf den Ergebnissen der ersten beiden Förderungsperioden auf [28][29]. Die im *Arbeitschwerpunkt A* zu entwickelnde Methodik basiert a) auf Erfahrungen bei der Modellierung von Methoden für die Entwicklung eingebetteter Systeme und der Ermittlung der generischen Parameter dieser Methoden [5][25][13][22], b) einer werkzeuggestützten Methodik zur Erstellung maßgeschneiderter Projektpläne [5][27][18], c) auf einem Schema zur Repräsentation von Messplänen sowie einer Methodik zu deren Erstellung [9][17][26][10], d) auf der Verfügbarkeit bereits existierender Wiederverwendungsartefakte (z. B.: Planfragmente, Qualitätsmodelle) in der Erfahrungsdatenbank des SFB [12]. Die in *Arbeitsschwerpunkt B* durchzuführenden Forschungsarbeiten basieren auf a) einer im Teilprojekt entwickelten Experimentiermethodik und der Mitarbeit bei der Planung, Durchführung und Analyse von Fallstudien und kontrollierten Experimenten [11][21][23][19][20], b) einer initialen Menge von Experimentfamilien [2]. Die in *Arbeitsschwerpunkt C* vorgesehenen Forschungsarbeiten basieren auf a) Erfahrungen im Bereich "Messen und Bewerten" [24] sowie b) der Entwicklung von Szenarien, die den Nutzen einer zielorientierten, projektbegleitenden Interpretation und Visualisierung von Messdaten in typischen Umlangungssituationen demonstrieren [14][15][16].

Literatur

- [9] Eleni Chaitidou: *Erstellung von Projekt- und GQM-Plänen für ein Software-Entwicklungsprojekt*. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.
- [10] Ralph Eiselein: *Definition of measurement points and measurement procedures by using process models and GQM plans*. Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.
- [11] Raimund L. Feldmann, Jürgen Münch, Stefan Queins, Stefan Vorwiegler und Gerhard Zimmermann: *Baselining a Domain-Specific Software Development Process*. Technischer Bericht Nr. 02/97, Sonderforschungsbereich 501, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.
- [12] Raimund L. Feldmann, Jürgen Münch und Stefan Vorwiegler: *Towards Goal-Oriented Organizational Learning: Representing and Maintaining Knowledge in an Experience Base*. In Proceedings of the 10th International Conference on Software Engineering and Knowledge Engineering (SEKE'98), San Francisco, USA, 18.-20. Juni 1998.
- [13] Birgit Geppert, Jürgen Münch und Erik Petersen: *Ein formales Prozeßmodell für die Software-Entwicklungsmethode SOMT (A Formal Process Model for the Software Development Method SOMT)*. Technischer Bericht Nr. 03/98, Sonderforschungsbereich 501, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.
- [14] Sigrid Goldmann, Jürgen Münch und Harald Holz: *A Meta-Model for Distributed Software Development*. In Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'99), Stanford, California, USA, 16.-18. Juni 1999.
- [15] Sigrid Goldmann, Jürgen Münch und Harald Holz: *Distributed Process Planning Support with MILOS*. In Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering (SEKE'99), Kaiserslautern, 17.-19. Juni 1999. Erweiterte Fassung in International Journal of Software Engineering and Knowledge Engineering, Oktober 2000.
- [16] Sigrid Goldmann, Jürgen Münch und Harald Holz: *Distributed Process Planning Support with MILOS*. In International Journal of Software Engineering and Knowledge Engineering, August 2000.
- [17] Matthias Gutheil: *Wartbare Schema-Repräsentation für GQM-Pläne*. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.
- [18] Jens Heidrich: *Conceptual Modeling of Software Development Processes*. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, Februar 2000.

- [19] Erik Kamsties, Antje von Knethen, Jan Philipps und Bernhard Schätz: *Eine vergleichende Fallstudie mit CASE-Werkzeugen für objektorientierte und funktionale Modellierungstechniken*. OMER Workshop, 28.-29. Mai 1999, P. Hofmann, A. Schürr (EDS.), OMER Workshop Proceedings Bericht Nr. 1999-01, Mai 1999.
- [20] Erik Kamsties, Antje von Knethen, Jan Philipps und Bernhard Schätz: *Eine vergleichende Fallstudie mit CASE-Werkzeugen für formale und semi-formale Beschreibungstechniken*. GI/ITG-Fachgespräch formale Beschreibungstechniken für verteilte Systeme, 1999.
- [21] Antje von Knethen, Raimund Feldmann, Jürgen Münch, Stefan Vorwieger: *A Baseline for the Development of Embedded Software*. Technischer Bericht Nr. 4/00, Sonderforschungsbereich 501, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 2000.
- [22] Antje von Knethen, Jürgen Münch: *Entwicklung eingebetteter Software mit UML: Der Do-it-Prozess V 1.0*, Technischer Bericht Nr. 05/00, Sonderforschungsbereich 501, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 2000.
- [23] Antje von Knethen, Erik Kamsties, Ralf Reussner und Bin Shen: *A Comparative Case Study with Industrial Requirements Engineering Methods*. 11th International Conference on Software Engineering and its Applications, Paris, 8.-10. Dezember 1998.
- [24] F. van Latum, R. van Solingen, M. Oivo, B. Hoisl, D. Rombach und G. Ruhe: *Adopting GQM-based Measurement in an Industrial Environment*. In IEEE Software, Seiten 78-86, Januar/Februar 1998.
- [25] Erik Petersen: *Formalisierung ausgewählter Software-Entwicklungsmethoden für reaktive Systeme in MVP-L (Formalization of selected software development methods for reactive systems)*. Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.
- [26] Eric Ras: *Export-/ Importschnittstelle für das Werkzeug GQMPlanner*. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.
- [27] Björn Schmidt: *Modellierung eines adaptierbaren Software-Entwicklungsprozesses*. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.
- [28] Sonderforschungsbereich 501: *Arbeits- und Ergebnisbericht 1995/1996/1997*. Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1997.

- [29] Sonderforschungsbereich 501: *Arbeits- und Ergebnisbericht 1998/1999/2000*. Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 2000.

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Langfristiges Ziel dieses Teilprojekts ist die Entwicklung einer generisch ausgelegten Experimentiermethodik zur Gewinnung empirischer Erfahrungen über Software-Entwicklungsprozesse und -ansätze (Techniken, Methoden, Werkzeuge). Dazu werden die folgenden drei Forschungsschwerpunkte bearbeitet:

Generierung umfassender Projektpläne

Bei der empirischen Erfahrungsgewinnung spielt neben der Prozesssicht die Messsicht eine wichtige Rolle. Die Prozesssicht enthält Informationen über den Entwicklungsprozess, zugeordnete Produkte und Ressourcen sowie Angaben über die Abwicklungsreihenfolge und Zeitvorgaben. Die Messsicht umfasst Informationen zur Definition, Erfassung, Analyse und Interpretation von Messdaten. Im Rahmen der Projektplanung sind darüber hinaus weitere Sichten wie beispielsweise die Konfigurationssicht zu berücksichtigen. In der ersten Förderungsperiode wurden im Teilprojekt B1 Notationen zur Beschreibung der Prozess- und Messsicht entwickelt und erprobt. In der zweiten Förderungsperiode wurden hierauf aufbauend Techniken entwickelt, die es ermöglichen, Prozess- und Messpläne mittels generischer Methoden zu erstellen. In diesem Kontext werden generische Methoden als Beschreibungs- und Generierungstechniken verstanden, durch die eine Wiederverwendung von Prozess- und Messwissen bei der Erstellung neuer Pläne methodisch unterstützt wird. Hierfür wurden Fragmente von Prozess- und Messplänen generisch ausgelegt, d. h., Teilaspekte wurden nur strukturell und damit hinsichtlich konkreter Ausprägung unvollständig spezifiziert. Zur vollständigen Spezifikation der Planfragmente müssen die generischen Parameter instantiiert werden. Die in der zweiten Förderungsperiode entwickelten generischen Methoden sehen vor, dass Prozess- und die Messpläne separat erstellt und anschließend integriert werden. Die Integration erfolgt mittels einer Konsistenzanalyse der beiden Sichten, einer Beseitigung von Widersprüchen und einer anschließenden Instrumentierung der Prozesssicht. Eine solche Vorgehensweise hat zunächst den Vorteil, dass Prozess- und Messaspekte separat untersucht und Variationen innerhalb der jeweiligen Sicht identifiziert werden können.

Folgende Hauptprobleme sind jedoch mit einer solchen Vorgehensweise verbunden:

- Fragmentierung von Sichten: Zusammengehörige Aspekte wie z. B. die Aggregation von Entwicklungsprozessen und die zugehörige Verdichtung von Messgrößen werden getrennt behandelt und teilweise redundant beschrieben. Hierbei können Inkonsistenzen entstehen, die behoben werden müssen. Diese Problematik ist vergleichbar mit der weitgehend separaten Modellierung von

Sichten bei der objektorientierten Softwareentwicklung mit UML (z. B.: der strukturellen Sicht in Form von Klassendiagrammen und der Verhaltenssicht in Form von Sequenz-, Zustands-, Aktivitäts- und Kollaborationsdiagrammen). Die Fragmentierung von Sichten bereitet bei der Projektplanung insbesondere dann Schwierigkeiten, wenn Teile des Plans generiert werden. Wechselseitige Abhängigkeiten zwischen verschiedenen Sichten (insbesondere zwischen der Mess- und der Prozesssicht) werden in diesem Falle nicht systematisch berücksichtigt.

- Unzureichende Berücksichtigung umfassender Planungsinformation: Neben den bisher berücksichtigten Sichten gibt es weitere Aspekte, die bisher separat behandelt werden (z. B. die Konfigurationsplanung). Eine solche Trennung führt unter anderem zu einer ineffizienten Planerstellung. Beispielsweise wird das Produktmodell redundant für die Konfigurationsplanung modelliert, obwohl es bereits Bestandteil der Prozesssicht ist. Weitergehende Konsequenzen einer auf Einzelaspekte konzentrierten Planung können eine zu geringe Kopplung zwischen verschiedenen Sichten sein und eine damit einhergehende Zweckentfremdung einzelner Sichten. Beispielsweise kann eine nicht ausreichend an den Prozess gekoppelte Konfigurationsplanung zur Folge haben, dass das Konfigurationsmanagement die (so nicht vorgesehene) Aufgabe der Projektsteuerung übernimmt.

Es wird erwartet, dass durch die Entwicklung einer Methodik zur Generierung integrierter Mess- und Prozesspläne und die darauf aufbauende Integration weiterer Planungssichten ein entscheidender Beitrag zur Lösung der erwähnten Probleme beigesteuert werden kann. Im Einzelnen ist Folgendes vorgesehen: Die Erstellungsprozesse für Mess- und Prozesspläne sollen miteinander verzahnt werden. Die Verzahnung soll schrittweise und parallel erfolgen, sodass stufenweise konsistente Resultate entstehen. Dies ermöglicht einerseits eine zielführende Auswahl geeigneter Entwicklungsansätze auf unterschiedlichen Abstraktionsebenen des Software-Entwicklungsprozesses und andererseits eine konsistente Instrumentierung der Prozesspläne. Erfahrungen des Teilprojekts haben gezeigt, dass die Generierung von Mess- und Prozessplänen die Verwendung expliziter Qualitätsmodelle erfordert. Solche Modelle beschreiben den Zusammenhang zwischen messbaren Einflussfaktoren und relevanten Qualitätsmerkmalen. Resultat eines derartigen Generierungsprozesses ist ein Projektplan, der die Prozess- und Messplanung mit expliziten Qualitätsmodellen integriert. In Zusammenarbeit mit Teilprojekt A1 sollen umfassende Projektpläne erstellt werden, die alle relevanten Planungsaspekte (insbesondere Prozessplanung, Kosten-, Zeit- und Personalplanung, Messplanung, Konfigurationsplanung, Qualitätssicherungsplanung, Projektmonitoring-Planung) umfassen. Aufgebaut wird a) auf in der ersten Förderperiode entwickelten Notationen zur expliziten Modellierung von Prozessen, b) auf in der zweiten Förderperiode entwickelten kompositions- und generierungsbasierten Methoden zur Wiederverwendung von Prozessmodellen, sowie c) einer Menge von Qualitätsmodellen, die in der Erfahrungsdatenbank des SFB vor-

handen ist. Zugrundeliegende Methoden und Techniken basieren auf den Prozessmodellierungsansätzen MVP-L und MILOS sowie dem GQM-Messansatz.

Im Kontext dieses Forschungsschwerpunkts existieren folgende Forschungsfragen:

- Welche Beziehungen existieren zwischen unterschiedlichen Plan-Sichten?
- Welcher Effizienzgewinn kann durch Generierung erreicht werden?

Generierung von Experimentfamilien

Die Erprobung von Modellen, Methoden, Techniken und Werkzeugen im SFB erfolgt mit Hilfe von Fallstudien und kontrollierten Experimenten, durch die sich systematisch Erkenntnisse gewinnen lassen. Dies erfordert eine Methodik zur Durchführung von Experimenten im Software Engineering im allgemeinen, die für den SFB instantiiert werden muss. Im Rahmen dieser Methodik können verschiedene Experimenttechniken (z. B.: Entwurfs-, Analyse-, Datenvalidierungstechniken) angewandt werden. In der ersten Förderungsperiode wurde initial eine experimentelle Methodik entwickelt, die in der zweiten Förderungsperiode anhand von Erfahrungen in durchgeführten Experimenten verfeinert bzw. modifiziert wurde. Die hierbei zu treffende Auswahl einzelner Experimentiertechniken (Entwurfs-, Datenvalidierungs-, Analyse-, Interpretationsverfahren etc.) erfolgte im jeweiligen Kontext der Experimente im Anschluss an eine gründliche Analyse der Einflussfaktoren. Insbesondere die Auswahl und Anpassung experimenteller Entwürfe, sogenannter *Experimentpläne*, ist abhängig von einer Vielzahl von Einflussfaktoren. Eine wichtige Rolle spielen dabei der untersuchte Gegenstand (Modell, Technik, Methode, Werkzeug), der zu untersuchende Qualitätsaspekt (z. B.: Zuverlässigkeit), der experimentelle Kontext (z. B.: Anzahl der Subjekte, zeitliche Restriktionen) sowie der mit dem Experiment verfolgte Zweck. Darüber hinaus haben die beabsichtigte Skalierbarkeit, die gewünschte Validität der Resultate und der geplante Grad der Wiederholbarkeit einen entscheidenden Einfluss auf die Auswahl und Anpassung der verwendeten Entwurfstechnik.

Beim Einsatz der experimentellen Methodik wurden folgende Hauptprobleme identifiziert:

- Ineffiziente Analyse der Einflussfaktoren: Ausgehend von den jeweiligen Vorbedingungen, die beim Einsatz von Experimentiertechniken gelten müssen, wurde im Sinne einer Checklistenüberprüfung eine Menge verschiedener Techniken hinsichtlich ihrer Anwendbarkeit bewertet und darauf basierend eine Auswahl getroffen. Hierbei wurde bereits eine domänenspezifische Vorauswahl berücksichtigt, jedoch nicht spezielle Randbedingungen, die bei Klassen von Experimenten gelten. Solche Klassen (sog. Familien) könnten z. B. Experimente mit Echtzeit-Spezifikationstechniken sein oder Experimente, in deren Mittelpunkt eine Untersuchung des Zuverlässigkeitsaspekts steht.

- Unzureichende Kenntnis der Abhängigkeiten zwischen Experimentiertechniken: Einzelne Experimentiertechniken können sich gegenseitig beeinflussen. So kann beispielsweise eine Entwurfstechnik nur in Kombination mit einer Menge von Analyseverfahren verwendet werden, die wiederum bestimmte Validitätsgrade der Resultate bedingen. Die bisher existierende experimentelle Methodik berücksichtigt solche Abhängigkeiten nur indirekt. Es fehlt eine explizite Darstellung der Abhängigkeiten.
- Geringe Wiederverwendbarkeit: Die Wiederverwendung von Experimentplänen, die in vergangenen Experimenten verwendet wurden, ist nur unzureichend möglich. Ein Grund hierfür ist das weitgehende Fehlen von Konstrukten zur Beschreibung generischer Experimentpläne.

Es wird erwartet, dass die Generierung von Experimentfamilien einen wesentlichen Beitrag zur Lösung der erwähnten Probleme darstellt. Hierbei kann auf existierenden Experimentfamilien aufgebaut werden [2]. Der Ausdruck "Generierung von Experimentfamilien" subsumiert die effektive und effiziente, wiederverwendungsorientierte Erstellung von maßgeschneiderten Plänen für die Durchführung von Experimenten (eingeschränkt auf Experimentklassen, die für den SFB eine herausragende Bedeutung haben). Metaanalysetechniken bieten die Möglichkeit, systematisch Ergebnisse ähnlicher Experimente zu analysieren und daraus Schlüsse zu ziehen, die im Kontext der Einzelexperimente nicht ermittelbar sind. Dieses analytische Wissen soll bei der Generierung von Experimentfamilien berücksichtigt werden. Insbesondere sollen hiermit Aussagen darüber gemacht werden, welche zusätzlichen Experimentvarianten bzw. welche Serie von Experimenten durchgeführt werden muss, um bestimmte experimentelle Hypothesen überprüfen zu können.

Im Einzelnen ist in diesem Forschungsschwerpunkt folgendes vorgesehen: Die in diesem Teilprojekt entwickelte Experimentiermethodik soll als anpassbarer Prozess zur empirischen Erfahrungsgewinnung formuliert werden. Dies beinhaltet die genaue Definition von Experimentzielen und -kontexten sowie hierauf aufbauend die geeignete Auswahl und Anpassung von Experimentplänen und Techniken zur Datenerfassung, -validierung, -analyse und -interpretation. Wesentliche Einflussfaktoren hinsichtlich der Beschaffenheit des Prozesses sind zu berücksichtigen. Die Kenntnis und explizite Darstellung des Einflusses solcher Faktoren auf verschiedene Experimentiertechniken unterstützt die effiziente Generierung einer maßgeschneiderten Vorgehensweise, verspricht die Erzielung mathematisch valider Resultate und ermöglicht die Generierung von Experimentvarianten für wichtige Klassen experimenteller Fragestellungen. Aufgebaut wird auf der im Teilprojekt entwickelten Experimentiermethodik und deren Erprobung/Optimierung im SFB sowie dem QIP-Ansatz für lernende Software-Entwicklungsorganisationen. Im Falle einer Durchführung von Industrieprojekten in der dritten Förderungsperiode wird angestrebt, auch die Durchführung von externen Experimenten mit der Methodik zu unterstützen.

In diesem Zusammenhang existieren folgende Forschungsfragen:

- Welches sind wichtige Einflussfaktoren für Experimente?
- Wie spezifiziert man Projektziele und -charakteristika für Experimente?
- Welche Reuse-Techniken (kompositions-/generierungsbasiert) sind geeignet?
- Wie integriert man interne und externe Erprobungsergebnisse?

Prozessleitstand

Die zielgerichtete Steuerung von Experimenten und Projekten setzt ein verteiltes Qualitätsmanagement voraus, das die plangerechte Abwicklung des Entwicklungsprozesses überwacht und Zielabweichungen (z. B.: Qualitätseinbußen, Verzögerungen) frühzeitig erkennt. Zu diesem Zwecke wird ein Prozessleitstand eingerichtet, mit dem folgende Ziele verfolgt werden: a) erhöhte Sichtbarkeit des Projektfortschritts durch den Vergleich von Soll-Ist-Daten, b) kontinuierliche Qualitätskontrolle, c) eine Erhöhung der Übereinstimmung der realen Entwicklungstätigkeiten mit dem Entwicklungsprozess, d) eine Verbesserung der Entscheidungsgrundlage bei Umplanungen, e) eine genauere Bewertung von Projekten zum Zwecke der Verbesserung. In den ersten beiden Förderungsperioden wurden Offline-Datenanalysen zur Projektkontrolle in Anwendungsprojekten des SFB durchgeführt.

Hierbei wurden folgende Hauptprobleme identifiziert:

- Fehlende Entscheidungsgrundlagen zur Projektsteuerung: Die Einleitung zielgerichteter Reaktionen (z. B. Umplanungen) bei Abweichungen von Qualitätszielen setzt eine systematische Qualitätskontrolle voraus. Die zielgerichtete Steuerung und Umplanung von Entwicklungsprojekten konnte nur bedingt durchgeführt werden, da nur wichtige Kenngrößen (insbesondere Fehler, Aufwand und Kalenderzeit) während des Projekts analysiert, interpretiert und dem Projektmanager zur Verfügung gestellt wurden. Somit mussten Umplanungsentscheidungen basierend auf wenigen quantitativen Kenngrößen sowie Projekt- und methodisch-bedingten Vorgaben getroffen werden. Erschwerend kam hinzu, dass die Aufbereitung der quantitativen Messdaten nicht rollenspezifisch erfolgte (z. B. für die Sicht eines Managers, eines Entwicklers oder eines Qualitätssicherers).
- Unzureichende Berücksichtigung der verteilten Entwicklung: Bisherige Ansätze des Qualitätsmanagements gehen typischerweise von einem zentralen, konzeptuell einheitlich geplanten und durchgeführten Entwicklungsprozess aus. Die zunehmende räumliche und zeitliche Verteilung von Software-Entwicklungstätigkeiten erfordert entsprechende Anpassungen des Qualitätsmanagements. Beispielsweise ist davon auszugehen, dass Messdaten an verschiedenen Orten in verschiedenen Formaten vorliegen. Neuere Qualitätsmanagementansätze müssen eine Integration über organisatorische Grenzen hinweg leisten, aktuelle Informationen über den Projektstatus allen

Projektbeteiligten zugänglich machen sowie die Analyse von Projektdaten in verteilten, heterogenen Umgebungen ermöglichen.

- Unzureichende Berücksichtigung der Spezifika der Software-Entwicklung: Viele existierende Monitoring-, Vorhersage- und Datenpräsentationsverfahren sind nicht an die Charakteristika der Software-Entwicklung angepasst. Solche Charakteristika sind insbesondere die Mensch-basierte Entwicklung in Teams, häufige Änderungen der Projektziele und des Projektkontexts oder die inhärente Kreativität vieler Entwicklungsschritte. Dies muss sich in den Anforderungen an einen Prozessleitstand widerspiegeln. Beispielsweise ist die Konsistenz subjektiver Maße zu gewährleisten, Vorhersagen sind unter Berücksichtigung des Projektkontexts zu ermitteln und Kontrollkriterien müssen änderbar sein.
- Geringer Automatisierungsgrad: Die Online-Analyse, -interpretation und -visualisierung von Messdaten wird derzeit nicht unterstützt. Die projektbegleitende Datenauswertung der wichtigsten Kenngrößen sowie die detaillierten Abschlussanalysen erfolgen derzeit (semi-)automatisch.

Es wird erwartet, dass durch die Untersuchung, Realisierung und Erprobung von Techniken zur verteilten, automatisierten Datenerfassung, -interpretation, -präsentation und Qualitätssicherung im Rahmen eines Prozessleitstands ein essentieller Beitrag zur Lösung der erwähnten Probleme beigesteuert werden kann. Im Einzelnen ist folgendes vorgesehen: Es sollen Techniken zur Einrichtung eines Prozessleitstands auf der Basis der projektspezifischen Erfahrungsdatenbank des SFB (integriert mit MILOS) realisiert und eingesetzt werden. Es soll ermöglicht werden, eine systematische Qualitätskontrolle während der Entwicklung durchzuführen, sodass begründete zielgerichtete Reaktionen (z. B. Umplanungen) bei Abweichungen von Qualitätszielen eingeleitet werden können.

Aufgebaut wird auf Erfahrungen mit der Datenanalyse zur Projektkontrolle, auf Projekttraces aus durchgeführten Projekten, dem Mechanismus zur Interpretation von Daten mittels GQM, einem von Teilprojekt A1 entwickelten Werkzeug zur prozessgekoppelten Produktverwaltung, das die Möglichkeit des Monitorings von Produktattributen anbietet, sowie der Ausführungsmaschine MILOS, die eine verteilte Entwicklung ermöglicht. Der SME- und der WebME-Ansatz (NASA Software Engineering Laboratory) zur Kombination und Modellierung verteilter Daten dienen als Ausgangspunkt für die Entwicklung geeigneter Analysetechniken des Prozessleitstands. Gegenwärtige angewandte Forschung im Kontext des VEKTOR-Projekts der Daimler-Chrysler AG mit dem Fraunhofer IESE unterstreicht die Praxisrelevanz dieses Forschungsschwerpunkts.

In diesem Forschungsschwerpunkt existieren folgende Forschungsfragen:

- Wie visualisiert man den Projektzustand in geeigneter Weise für Projektmanagement und Qualitätssicherung?

- Wie können neue Projekte mit vorangegangenen Projekten verglichen werden, um Trends und Abweichungen von erwartetem Verhalten zu bestimmen?
- Wie kann die Detektion von kritischen Projektzuständen (im Sinne eines “Frühwarnsystem”) automatisiert werden?

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Generierung umfassender Projektpläne

In diesem Arbeitsschwerpunkt wird eine Methodik zur Generierung umfassender Projektpläne unter Einbeziehung expliziter Qualitätsmodelle entwickelt. Dabei wird auf Vorarbeiten aus den ersten beiden Förderungsperioden aufgesetzt. Folgende Arbeitspakete sind vorgesehen:

Arbeitspaket A1: Entwicklung einer Methodik zur Generierung umfassender Pläne aus Projektzielen und -charakteristika

In diesem Arbeitspaket wird eine systematische Methodik zur Generierung von integrierten Mess- und Prozessplänen entwickelt. Eingabe für den Generierungsprozess ist eine detaillierte Beschreibung der Projektziele und -charakteristika sowie eine Menge von wiederverwendbaren Planfragmenten und Qualitätsmodellen. Die Methodik soll aufgrund dieser Eingabewerte maßgeschneiderte Pläne erzeugen. Zusätzlich ist eine geeignete Darstellungsform für die Beschreibung von Qualitätsmodellen zu entwickeln. Hinsichtlich der Wiederverwendung von Planfragmenten werden zwei prinzipielle Ansätze, die auch im Produktbereich Anwendung finden, unterschieden: 1) Beim kompositionsbasierten Ansatz werden durch das Zusammenfügen von Bauteilen Gesamtsysteme erstellt. Die Bauteile werden dabei mit möglichst wenigen Veränderungen aus früheren Entwicklungen übernommen. 2) Beim generierungsbasierten Ansatz wird aus einer Spezifikation halb- oder vollautomatisch ein Gesamtsystem erstellt. Dabei werden nicht Bausteine, sondern vielmehr typische Strukturen, Verwendungsmuster und Architekturen wiederverwendet. Für die Generierung eines maßgeschneiderten Prozessplans wurde im Teilprojekt bereits ein Ansatz entwickelt, der zunächst eine zielführende Planung mittels Komposition von Planfragmenten vorsieht. Die ausgewählten Planfragmente werden daraufhin mittels einer generierungsbasierten Methodik an den Projektkontext angepasst und schließlich integriert. Für Messpläne existiert bereits eine kompositionsbasierte Methodik, die eine Template-orientierte Wiederverwendung von Messplänen unterstützt. Aufbauend auf Erfahrungen dieser bereits vorhandenen Ansätze soll ein integrierter Ansatz entwickelt werden, der beide Sichten berücksichtigt.

Personalbedarf: 18 PM

Arbeitspaket A2: Integration weiterer Plansichten

In diesem Arbeitspaket wird in Zusammenarbeit mit A1 eine Methodik entwickelt, die es erlaubt, weitere Plansichten mit dem Prozess- und Messplan zu integrieren.

Dies sind insbesondere die Konfigurationsplanung, die Qualitätssicherungsplanung und die Projektmonitoring-Planung. Basierend auf existierenden Notationen zur Beschreibung dieser Planungsaspekte sind Kriterien für eine geeignete Darstellung der Zusammenhänge zwischen den verschiedenen Sichten abzuleiten (z. B.: Koppelung von Produkt- und Konfigurationsmodell). Anhand dieser Kriterien sollen existierende Notationen untersucht und mit den im Teilprojekt bereits entwickelten Notationen integriert werden. Dies setzt eine sorgfältige Analyse der wechselseitigen Abhängigkeiten zwischen den einzelnen Sichten voraus.

Personalbedarf: 12 PM

Arbeitsschwerpunkt B: Generierung von Experimentfamilien

In diesem Arbeitsschwerpunkt wird eine Methodik zur Generierung von Experimentfamilien entwickelt. Dabei wird auf Vorarbeiten aus den ersten beiden Förderungsperioden aufgesetzt. Als externer Berater für diesen Arbeitsschwerpunkt konnte Professor Basili von der University of Maryland, USA, gewonnen werden. Folgende Arbeitspakete sind vorgesehen:

Arbeitspaket B1: Analyse relevanter Einflussfaktoren von Experimentplänen und Experimentiertechniken

In diesem Arbeitspaket werden Einflussfaktoren identifiziert, die die Auswahl und Anpassung von Experimentiertechniken (speziell: Entwurfs-, Datenvalidierungs und Analysetechniken) beeinflussen. Des Weiteren werden wechselseitige Abhängigkeiten zwischen verschiedenen Experimentiertechniken und deren Kombinierbarkeit untersucht und explizit beschrieben. Beispielsweise muss berücksichtigt werden, dass die Verwendung einiger Analysemethoden nur bestimmte Arten von Experimentplänen und Datenvalidierungstechniken erlaubt, die wiederum nur in bestimmten Kontexten anwendbar sind. Die Ausprägung dieser Parameter ist entscheidend für die Planung des Experiments. Die Wirkung der identifizierten Parameter auf Experimentiertechniken wird untersucht und beschrieben.

Personalbedarf: 6 PM

Arbeitspaket B2: Generische Auslegung Experimentplänen

In diesem Arbeitspaket sollen ausgewählte Experimentpläne generisch ausgelegt werden. Der Entwurf ist der zentrale Bestandteil eines Experiments und hat großen Einfluss auf die praktische Realisierung des Experiments. Experimentpläne (z. B.: Zwei-Gruppen-Plan, fraktional-faktorieller Plan) sind für Klassen von Experimenten, die im SFB eine herausragende Rolle spielen, auszuwählen und generisch auszulegen. Dies umfasst die Entwicklung von Konstrukten zur Beschreibung generischer Experimentpläne. Des Weiteren sind Informationen über die Konsequenzen der Auswahl eines bestimmten Entwurfstyps zu ermitteln und explizit zu beschreiben.

Personalbedarf: 6 PM

Arbeitspaket B3: Generierung von Experimentvarianten

In diesem Arbeitspaket soll die in diesem Teilprojekt entwickelte Experimentiermethodik als anpassbarer Prozess zur empirischen Erfahrungsgewinnung formuliert werden. Dies beinhaltet die genaue Definition von Experimentzielen und -kontexten sowie hierauf aufbauend die geeignete Auswahl und Anpassung der generisch ausgelegten Experimentiertechniken. Ziel des Arbeitspaketes ist es, für ausgewählte Experimenttypen effektiv und effizient eine Anleitung erstellen zu können, mittels derer Experimente korrekt durchgeführt und mathematisch valide Resultate erzielt werden können.

Personalbedarf: 10 PM

Arbeitspaket B4: Vergleich von SFB-internen mit SFB-externen Erprobungsergebnissen

Im Falle einer Durchführung von Industrieprojekten in der dritten Förderungsperiode wird angestrebt, auch die Durchführung von externen Experimenten mit der Methodik zu unterstützen. Hierbei sind besondere Rahmenbedingungen zu berücksichtigen, die im industriellen Kontext existieren (z. B.: geringe Verfügbarkeit professioneller Entwickler; Parallelität oder Überdeckung von Fallstudien/Experimenten mit industriellen Anwendungsentwicklungen). Des Weiteren ist geplant, SFB-interne und SFB-externe Experimentergebnisse unter Berücksichtigung der Verschiedenartigkeit der Kontexte miteinander zu vergleichen. Hierfür ist der Einsatz von Metaanalyseverfahren geplant.

Personalbedarf: 9 PM

Arbeitsschwerpunkt C: Prozessleitstand

In diesem Arbeitsschwerpunkt erfolgt die Untersuchung, Realisierung und Erprobung von Techniken zur automatisierten Datenerfassung, -interpretation, -präsentation und Qualitätssicherung im Rahmen eines Prozessleitstands. Dabei wird auf Vorarbeiten aus den ersten beiden Förderungsperioden aufgesetzt. Folgende Arbeitspakete sind vorgesehen:

Arbeitspaket C1: Entwicklung einer Technik zur zielorientierten Erfassung, Interpretation und Präsentation von Messdaten

In diesem Arbeitspaket soll eine umfassende Übersicht über Techniken zur Kontrolle, Analyse, Interpretation und Visualisierung von Messdaten erstellt werden. Hierunter fallen insbesondere Techniken aus den Bereichen Monitoring, Controlling sowie der Datenpräsentation. Gegeben werden soll eine Übersicht der Techniken im Bereich der Software-Entwicklung sowie in anderen Anwendungsfeldern. Hierbei ist insbesondere gedacht an a) wirtschaftswissenschaftliche Anwendungsfelder (Geschäftsprozesse, Produktionsprozesse, Entwicklungsprozesse) sowie b) ingenieurwissenschaftliche Anwendungsfelder (physikalische Prozesse; z. B. Überwachung von Regelkreisen). Des Weiteren sollen Anforderungen an einen Prozessleitstand in der Software-Entwicklung aufgestellt werden. Hierbei sind die Spezifika

der Software-Entwicklung (z. B. Mensch-basierte Entwicklung, Erfordernis der Anpassung von Techniken an unterschiedliche Kontexte) zu berücksichtigen. Die untersuchten Techniken sollen daraufhin anhand der ermittelten Anforderungen bewertet werden. Hierauf aufbauend soll eine Methode zur zielorientierten Kontrolle, Analyse, Interpretation und Visualisierung des aktuellen Projektzustandes entwickelt werden. Dafür sollen geeignete Techniken ausgewählt, angepasst und integriert werden. Des Weiteren soll auf dem GQM- und dem MVP- beziehungsweise MILOS-Ansatz aufgesetzt werden.

Personalbedarf: 18 PM

Arbeitspaket C2: Realisierung eines unterstützenden Werkzeugs zur prozessbegleitenden Interpretation und Visualisierung von Messdaten

In diesem Arbeitspaket soll ein Werkzeugs realisiert werden, dass die in Arbeitspaket C1 beschriebene Methode teil-/automatisiert. Insbesondere soll es die zur Qualitätskontrolle notwendigen Informationen für einen Qualitätssicherer visualisieren und die Messdaten als Grundlage für die Projektsteuerung geeignet aufbereiten (für die Rolle des Projektmanagers). Eine Anbindung an die flexible Ausführungsmaschine MILOS ist vorgesehen. Für die Realisierung des Prozessleitstandes sind in Abstimmung mit Teilprojekt A2 folgende Inkremente vorgesehen: 1) Erstellung eines Systementwurfs gemäß den in Arbeitspaket C1 abgeleiteten Anforderungen; 2) Entwicklung einer Analysekomponente zur prozessbegleitenden Analyse und Interpretation der Messdaten; 3) Entwicklung einer Visualisierungskomponente zur zweckorientierten, rollenspezifischen Darstellung der Informationen im MILOS-Planungs- und Qualitätssicherungsfenster. Die Validierung des Werkzeugs erfolgt in Arbeitspaket D3.

Personalbedarf: 21 PM

Arbeitsschwerpunkt D: Evaluierung entwickelter Techniken und Werkzeuge

In diesem Arbeitsschwerpunkt werden die Methoden zur Generierung von umfassenden Projektplänen, Experimentfamilien sowie der Prozessleitstand im Rahmen von Fallstudien und Experimenten empirisch validiert. Dazu wird insbesondere das SFB-Praktikum herangezogen. Folgende Arbeitspakete sind vorgesehen:

Arbeitspaket D1: Evaluierung der Methodik zur Generierung von umfassenden Projektplänen

In diesem Arbeitspaket wird eine Evaluierung der in Arbeitsschwerpunkt A entwickelten Methodik im Rahmen von Experimenten und Fallstudien durchgeführt. Ein wesentliches Ziel der Evaluierung ist der Nachweis, dass mit Hilfe der Methodik und des Werkzeugs umfassende Projektpläne mit weniger Aufwand und mit weniger Inkonsistenzen generiert werden können als ohne diese Methodik.

Personalbedarf: 6 PM

Arbeitspaket D2: Evaluierung der Methodik zur Generierung von Experimentfamilien

In diesem Arbeitspaket wird die in Arbeitsschwerpunkt B entwickelte Methodik zur Generierung von Experimentfamilien erprobt. Es soll die Fragestellung überprüft werden, ob der Grad der Wiederverwendung von Experimentplänen im Vergleich zur bisherigen Vorgehensweise signifikant gesteigert werden kann und dies mit einer Aufwandsreduktion bei der Planung von Experimenten im Vergleich zum traditionellen Aufsetzen von Experimenten verbunden ist. Unter “traditionelles Aufsetzen von Experimenten” wird hier die Checklisten-basierte Auswahl und manuelle Anpassung von Experimentiertechniken entsprechend vorgegebener Experimentziele verstanden. Eine weitere Beurteilung der Eignung der Methodik im Hinblick auf eine vollständige Erfassung der Abhängigkeiten soll durch qualitative Analysen erfolgen.

Personalbedarf: 6 PM

Arbeitspaket D3: Evaluierung des Prozessleitstands

In diesem Arbeitspaket wird der in Arbeitsschwerpunkt C entwickelte Prozessleitstand hinsichtlich seiner Eignung als Frühwarnsystem und Projektmonitoring-System bewertet. Durch den Einsatz des Prozessleitstandes und des unterstützenden Werkzeugs wird eine signifikante Steigerung des Automatisierungsgrades und eine verbesserte Entscheidungsunterstützung erwartet.

Personalbedarf: 6 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausrüstung Marcus Ciolkowski 1/2 N.N.						
	B1		B2	B3		
					D2	
	B4					
Ergänzungsausrüstung Jürgen Münch* N.N. N.N., stud. Hilfskraft N.N., stud. Hilfskraft						
	A1			A2		
					D1	
	C1					
			C2		D3	
	C1					
				D1 und D2		
			C2		D3	

* Die Aufnahme von Dipl.-Inform. Jürgen Münch als Projektleiter ist nach Abschluss seiner Dissertation vorgesehen. Für die Bearbeitung der Arbeitspakete sind 28,5 Std. / Woche vorgesehen, für die Projektleitung 10 Std. / Woche.

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Die Durchführung von Anwendungsentwicklungen und Experimenten erfolgt unter Anleitung des SE-Labors. Hierfür werden die Ergebnisse des Teilprojekts B1 in Form von Methoden und Prototypwerkzeugen in das SE-Labor integriert und stehen damit dem gesamten SFB zur Verfügung. Die gewonnenen Erfahrungen bilden die Grundlage für Weiterentwicklungen in Teilprojekt B1.

Die D-Teilprojekte stellen das notwendige Anwendungswissen zur Identifikation relevanter Variationsparameter für eine generische Auslegung von umfassenden Projektplänen und Experimentvarianten des Anwendungsgebiets „Gebäudeautomation“ bereit.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	1	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
			StHkr.	2	40.800	StHkr.	2	40.800	StHkr.	2	40.800
	zus.:	1	zus.:	2	249.600	zus.:	2	249.600	zus.:	2	249.600
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		3.000	(522)		3.000	(522)		3.000
			zusammen		3.000	zusammen		3.000	zusammen		3.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Rombach, D.; Prof. Dr. (2) Ciolkowski, M.; Dipl.-Inform. (3) 1/2 N.N. (4) Basili, V. R.; Prof. Dr. (5) Dellen, B.; Dr.	Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL Univ. of Maryland Fraunhofer IESE	5 20 10 B B	1.1.95 1.7.99 1.1.01	
Ergänzungsausrüstung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Münch, J.; Dipl.-Inform. (2) N.N. (3) N.N., StdHkr. (4) N.N., StdHkr.	Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	38,5 38,5 19 19	15.3.95 1.1.01 1.1.01 1.1.01	Ila Ila StHkr StHkr
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Die erste wissenschaftliche Mitarbeiterstelle der Grundausrüstung führt die konzeptionellen Arbeiten in dem Bereich “Generierung von Experimentfamilien” durch (Arbeitspakete B1-B3, D2).

Die zweite (halbe) wissenschaftliche Mitarbeiterstelle der Grundausrüstung entwickelt Verfahren zum Vergleich industrieller und SFB-interner Experimentresultate und wendet diese an (Arbeitspaket B4) .

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Die erste wissenschaftliche Mitarbeiterstelle der Ergänzungsausrüstung führt die zentralen konzeptionellen Arbeiten in dem Bereich “Generierung umfassender Projektpläne” durch (Arbeitspakete A1, A2, D1). Der Schwerpunkt liegt somit auf der *Planung* von Projekten und Experimenten. Für diese wissenschaftliche Mitarbeiterstelle ist Dipl.-Inform. Jürgen Münch vorgesehen, der nach Abschluss seiner Dissertation Leitungsaufgaben im Teilprojekt B1 übernehmen soll.

Die zweite wissenschaftliche Mitarbeiterstelle führt die zentralen konzeptionellen Arbeiten im Bereich “Prozessleitstand” durch und ist für dessen Realisierung verantwortlich (Arbeitspakete C1, C2, D3). Der Schwerpunkt liegt somit auf der *Abwicklung* von Projekten und Experimenten.

Die erste wissenschaftliche Hilfskraftstelle ist für die Bearbeitung der Arbeitspakete C1, D1 und D2 vorgesehen. Sie dient insbesondere der Unterstützung der umfangreichen Analysearbeiten sowie der vorgesehenen Evaluierungstätigkeiten.

Die zweite studentische Hilfskraftstelle ist für die Bearbeitung der Arbeitspakete C2 und D3 vorgesehen. Sie dient insbesondere der Unterstützung der umfangreichen Implementierungsarbeiten sowie der vorgesehenen Evaluierungstätigkeiten.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausstattung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausstattung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausstattung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausstattung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausstattung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
	-	-	-
Summe	-	-	-

3.1 Allgemeine Angaben zum Teilprojekt B2

3.1.1 Thema

Flexible Planung und Steuerung von Softwareentwicklungsprozessen

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Wissensbasierte Systeme und Künstliche Intelligenz

3.1.3 Leiter

Prof. Dr. Michael Richter
geb. 21.6.1938
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2800
Fax.: 0631/205-3357
richter@informatik.uni-kl.de

Die Stelle des Leiters ist befristet ☒ (x) nein ☐ () ja, bis zum

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ () ja	☒ (x) nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ () ja	☒ (x) nein
Tierversuche	☐ () ja	☒ (x) nein
gentechnologische Untersuchungen	☐ () ja	☒ (x) nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	645,2	6,0	-	651,2
1998	201,6	3,0	-	204,6
1999	206,4	3,0	-	209,4
2000	208,8	3,0	-	211,8
Zwischensumme	1262,0	15,0		1.277,0
2001	249,6	3,0	-	252,6
2002	249,6	3,0	-	252,6
2003	249,6	3,0	-	252,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Langfristiges Ziel des Teilprojekts B2 ist die Entwicklung von Methoden und Techniken zur flexiblen Planung und Steuerung komplexer Software-Entwicklungsprozesse. Hierbei liegt der Hauptaugenmerk darauf, Aktivitäten, die sich mit dem Management des Softwareentwicklungsprozesses beschäftigen (also Meta-Aktivitäten sind), möglichst fließend in die Projektdurchführung zu integrieren. In den ersten beiden Förderungsperioden wurden hierzu Techniken zur flexiblen Abwicklungsunterstützung und zur Unterstützung von Planverfeinerungen und Planänderungen während der Abwicklung realisiert. In Zusammenarbeit mit dem Teilprojekt B1 wurde die Software-Prozess-Modellierungssprache MILOS¹ entwickelt und auf deren Basis in der zweiten Förderungsperiode in Zusammenarbeit mit dem Teilprojekt A2 ein gleichnamiges Workflow-Management-System implementiert. Das MILOS-System erlaubt die Modellierung und Planung von Software-Entwicklungsprojekten und unterstützt ihre Durchführung u.a. mittels To-Do-Listen und Änderungsnotifikationen. Planverfeinerungen und Planänderungen während der Projektdurchführung werden unterstützt und gehen in die dem MILOS-System zugrundeliegende Abhängigkeitsverwaltung ein.

In der dritten Förderungsperiode soll nun die Planung selbst zum Gegenstand der Betrachtung und formalen Behandlung werden, beschrieben durch die Metaplanung. Damit wären alle drei für das B2-Projekt wesentlichen Aspekte (Modellierung, Planung und Ausführung) abgehandelt. Es ist notwendig, hierfür auch Verteilungsaspekte zu behandeln, also Techniken zur Unterstützung von verteilten SE-Prozessen mit zu berücksichtigen. Dabei stehen die resultierenden neuen konzeptionellen Fragestellungen, insbesondere die der verteilten Planung, im Mittelpunkt des Interesses.

Die Ergebnisse dieses Teilprojektes sollen prototypisch in MILOS integriert werden. Grundlage hierfür sind die konzeptionellen Vereinfachungen der Struktur von CoMo-

1. Modeling Language and Operational Support for Software Processes

Kit (Ergebnis der ersten Förderungsperiode), die in der letzten Periode im Rahmen der Teilprojekte A2 und B2 im Hinblick auf das langfristige Ziel der Verteilung bereits vorgenommen wurden.

Am Ende der dritten Förderungsperiode wird dann mit MILOS ein System zur Verfügung stehen, das eine flexible und verteilte Durchführung des gesamten Vorgangs von der ersten Problemdefinition über die Erstellung eines Prozessmodells und Projektplans bis hin zur Abwicklung (inklusive Umplanungen während der Abwicklung) unterstützt.

3.3 Stand der Forschung

Der Stand der Forschung wird hier nur bezüglich der Ergebnisse der letzten drei Jahre diskutiert.

Im Bereich der Metaplanung für Software-Engineering gibt es nur wenige Ansätze. Spade [1] besitzt die Mächtigkeit, Metapläne zu modellieren und auszuführen. Der Planer erhält aber keinerlei Unterstützung für dieses Vorgehen, z.B. existieren keine Default-Metapläne für bestimmte Planzustände, da es sich hierbei nicht um die Zielsetzung von Spade handelt.

Im Bereich der verteilten Abwicklung wurden in den letzten Jahren große Fortschritte gemacht. Verteilte Workflowsysteme wie Oz [7] und Endeavours [4] decken bereits einen großen Teil der Anforderungen an eine verteilte Abwicklung ab. Beide Systeme haben jedoch Schwächen. So stellt Endeavours zwar Unterstützung für dynamische Prozessänderungen zur Verfügung, die meisten Änderungen jedoch sind manuell vom Benutzer durchzuführen. Bei Oz liegt der Forschungsschwerpunkt auf der Ausführungsunterstützung, daher sind Umplanungen zwar möglich, werden aber vom System nur wenig unterstützt.

In den letzten Jahren zeichnete sich ein Wechsel von der prozesszentrierten hin zur artefaktzentrierten Sichtweise von Vorgängen ab. Der Vorteil dieser artefaktzentrierten Systeme liegt in einer Erhöhung der Flexibilität und Benutzerakzeptanz, da der Benutzer das zugrundeliegende transparente Prozeßmodell nicht explizit kennen muß, sondern es durch die Aktivitäten, die er auf den Artefakten durchführt, befolgt und auch ändern kann. Das PROSYT-System [2] beispielsweise ist ein artefaktzentriertes, verteiltes Softwareprozess-Unterstützungssystem.

Im Bereich der dynamischen Teambildung gibt es Ansätze im Bereich des Zusammenarbeits verschiedener (Entwicklungs-)Umgebungen [3]. Das Hauptaugenmerk liegt jedoch hier mehr auf der Kommunikation und Interaktion zwischen Firmen, nicht auf der Unterstützung und der Koordination eines gemeinsamen Softwareentwicklungsprozesses. Das Thema der globalen Softwareentwicklung wird beispielsweise in [5] behandelt. Hier liegt der Schwerpunkt ebenfalls auf der Zusammenarbeit zwischen den beteiligten Unternehmen und den hierbei aufkommenden Problemen. Zwar wird hier verdeutlicht, daß IT-Technologien in hohem Maße vorhanden sind und

genutzt werden sollen, aber es wird in zu geringem Maße darauf eingegangen, wie diese Technologien effizient genutzt werden können.

Zur Unterstützung der Teambildung mittels Softwareagenten sind Verhandlungstechniken und Strategien gefragt. Hier stellen Realisierungen und Erweiterungen des Contract-Net Protocols [6] einen interessanten Ansatzpunkt dar.

Einen Ausblick auf die zukünftige Entwicklung der Prozesssprachen gibt [8]. Hierbei sind Überschneidungen mit unseren Forschungsvorhaben und unserer derzeitigen Forschung auffällig.

- [1] G.Cugola: Tolerating Deviations in Process Support Systems via Flexible Enactment of Process Models. In: IEEE Transactions of Software Engineering, 24(11), Nov.1998.
- [2] G. Cugola, C. Ghezzi: Design and Implementation of PROSYT: A Distributed Process Support System; in IEEE Proceedings of the 8th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, USA, 1999.
- [3] G.E. Kaiser, St.E. Dossick, W. Jiang, J.Jingshuang Yang, S.X. Ye: WWW-based Collaboration Environments with Distributed Tool Services. In: World Wide Web Journal, Baltzer Science Publishers, 1998.
- [4] P.J. Kammer, G.A. Bolcer, R.N. Taylor, A.S. Hitomi: Supporting Distributed Workflow Using HTTP. ICSP 5 - 5th International Conference on Software Process Computer Supported Organizational Work, Chicago, USA, Juni 1998.
- [5] D.W. Karolak. Global Software development. IEEE Computer Society, 1998.
- [6] T. Sandholm, V. Lesser. Issues in automated Negotiation and Electronic Commerce: Extending the Contract Net Framework. Readings in Agents, 1998.
- [7] I.Z. Ben-Shaul, G.E. Kaiser: Federating Process-Centered Environments: the Oz Experience. In: Journal of Automated Software Engineering, 5(1), 1998.
- [8] M.S. Sutton, J.L.Osterweil. The Design of a Next-Generation Process Language. In Proceeding of the 5th ACM SIGSOFT Symposium on the Foundations of Software Engineering, No.1301 in Lecture Notes in Computer Science, Seiten142-158, Zürich, Schweiz, Springer-Verlag, Sep. 1997.

3.4 Eigene Vorarbeiten

Hier wird im wesentlichen auf den Ergebnisbericht des Teilprojekts B2 verwiesen. Einzig auf relevante Vorarbeiten hinsichtlich der neuen Aspekte soll noch eingegangen werden.

Um die Akzeptanz einer Software-Entwicklungsumgebung zu erhöhen ist es notwendig, dem Benutzer auch in einer global verteilten Entwicklungsumgebung den Zugriff auf ihm vertraute Arbeitswerkzeuge zu ermöglichen. So wurde in MILOS unter anderem das Planungswerkzeug Microsoft Project an MILOS angebunden. Diese Anbindung und weitere Funktionalitäten wurden auf der International Conference on Software Engineering (ICSE) 1999 im Rahmen einer System-Demonstration durchgeführt [12]. Eine Voraussetzung für den Erfolg dieser Demonstration war die verteilte globale Softwareentwicklung, die durch die Kooperation mit der Arbeitsgruppe von Professor Maurer an der Universität Calgary, Kanada, ermöglicht wurde [10].

Bei den Vorarbeiten zur Meta-Planung wurde zunächst Meta-Planung im Bereich des Mechanical Engineering untersucht [13], und das Wissen über die Techniken und Methodiken anschließend auf den Bereich des Software Engineering übertragen [14].

Bei der Unterstützung der verteilten Softwareentwicklung muss insbesondere auch auf die Aspekte der verteilten Teambildung eingegangen werden, wie sie z.B. bei der Bildung von virtuellen Unternehmen von Interesse sind. Hierzu wurden in [9] und [11] bereits erste Ideen erarbeitet.

- [9] B. Kötting, F. Maurer: A Concept for Supporting the Formation of Virtual Corporations through Negotiation, Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '99), IEEE Computer Society Press, 1999 (to appear).
- [10] F. Maurer, B. Dellen, F. Bendeck, S. Goldmann, H. Holz, B. Kötting, M. Schaaf: Merging project planning and web-enabled dynamic workflow for software development. In IEEE Internet Computing, Special Issue on Internet-Based Workflow - May/June 2000.
- [11] F. Maurer, B. Dellen, H. Holz: Process Support for Virtual Software Organizations, Proceedings of the 11th International Software & Engineering and Knowledge Engineering Conference (SEKE '99), Knowledge Systems Institute, ISBN 1-891706-01-2, 1999.
- [12] F. Maurer, G. Succi, H. Holz, B. Kötting, S. Goldmann, B. Dellen: Software Process Support over the Internet. Proceedings of the ICSE '99, Los Angeles, USA, 1999.
- [13] Ch. Petrie, S. Goldmann, A. Raquet: Agent-Based Project Management. In: Springer, in Lecture Notes in AI - 1600, Springer-Verlag, 1999.
- [14] S. Goldmann, J. Münch, H. Holz: Distributed Process Planning Support with MILOS. Proceedings of the SEKE '99, Kaiserslautern, 1999.

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Während in der ersten Förderungsperiode im Teilprojekt B2 Konzepte und Techniken zur Modellierung und Operationalisierung von Projektplänen und zur Verwaltung von Abhängigkeiten während der Projektdurchführung untersucht wurden, war das Forschungsthema der zweiten Förderungsperiode die Verwaltung von Abhängigkeiten zwischen Planung und Ausführung mit dem Ziel, die Plandurchführung flexibel an Änderungen des Plans anpassen zu können. Aufbauend auf diesen Ergebnissen soll der Forschungsschwerpunkt in der dritten Förderungsperiode jetzt auf der Unterstützung der Planung selbst liegen. Dabei sollen sowohl die initiale Planung als auch Verfeinerungen und Änderungen des Plans während der Projektdurchführung betrachtet werden. Hierbei werden wir die folgenden Techniken und Methoden einsetzen:

Meta-Modellierung

Ähnlich wie die Software-Entwicklung selbst ist der Vorgang der Planung von Software-Entwicklungsprojekten ein kreativer Entwurfsprozess und kann daher mit ähnlichen Mitteln modelliert werden wie diese. Da es sich hier um die Erstellung von Modellen zur Modellierung und Planung von (Softwareentwicklungs-) Prozessen handelt, sprechen wir im folgenden von *Meta-Modellierung* bzw. der Erstellung von *Meta-Plänen*. In einem Meta-Plan tauchen z.B. Aufgaben zur Zeitplanung oder Ressourcen-Zuordnung für Prozesse des Plans auf. Andere Meta-Aufgaben beschreiben die Notwendigkeit, den Plan an einer bestimmten Stelle durch Einfügen neuer Aufgaben zu verfeinern. Wie zwischen den Prozessen des Plans bestehen auch zwischen diesen Meta-Aufgaben Abhängigkeiten, die modelliert und vom System verwaltet werden sollen.

Ein Ziel der dritten Förderungsperiode im Teilprojekt B2 ist die durchgehende Modellierung (und Meta-Modellierung) des gesamten Prozesses der Software-Erstellung, von der Prozess-Modellierung und initialen Planung über die Projektdurchführung bis hin zur Auslieferung des fertigen Software-Produktes in einer einheitlichen Modellierungssprache. Ein so modelliertes Projekt kann dann mit Hilfe des in der zweiten Förderungsperiode in den Teilprojekten A2 und B2 entwickelten Prozess-Unterstützungswerkzeugs MILOS durchgängig von der initialen Planung bis zur Fertigstellung des auszuliefernden Produkts unterstützt werden. Dazu muss die in den ersten beiden Förderungsperioden in Zusammenarbeit mit den Teilprojekten B1 und A2 entwickelte Prozessmodellierungssprache so erweitert werden, dass die Besonderheiten der Planung berücksichtigt werden. Z.B. muss in Betracht gezogen werden, dass Planungsaufgaben (im folgenden auch Meta-Aufgaben genannt) keine (Software-) Produkte erzeugen, sondern Eigenschaften des Plans, bzw. der darin enthaltenen Prozesse, verändern. Desweiteren müssen Regeln und Integritätsbedingungen formalisiert werden, die einerseits der Aufdeckung inkonsistenter Ausprägungen des Plans dienen, andererseits in Verbindung mit geeigneten Heuristiken aber auch die automatische Generierung von Meta-Aufgaben unterstützen, mit dem Ziel, den Aufwand der (Meta-)

Prozessmodellierung in vertretbarem Rahmen zu halten. Die Wahl einer zu diesem Zweck adäquaten logischen Beschreibungssprache und die Verwaltung dieses Wissens kann sinnvollerweise in Kooperation mit Teilprojekt A2 erfolgen, welches in der nächsten Förderungsperiode Techniken des Wissensmanagements in MILOS integrieren soll.

Erstellung und Wiederverwendung von Plan- und Ausführungstraces

Änderungen während der Planung und Durchführung von Softwareentwicklungsprojekten sowie ihre Begründungen und Konsequenzen für das Projekt, sind oft von großem Interesse, um nachträglich Kosten- oder Zeitüberschreitungen zu analysieren und Fehler in der Planung und Durchführung zu identifizieren, damit sie in zukünftigen Projekten vermieden werden können. Zusätzlich zu diesem Aspekt der Wiederverwendung von Plan- und Ausführungstraces können vergangene Entscheidungen und ihre Begründungen sich auch im laufenden Projekt auf weitere Planungs- oder Ausführungsentscheidungen auswirken. In der zweiten Förderungsperiode wurden bereits Entscheidungen und Entscheidungsbegründungen (*Design Rationales*) während der Ausführung untersucht und deren Anwendung zur Analyse der Konsequenzen von (Plan- oder Produkt-) Änderungen während der Ausführung erforscht (*Change Impact Analysis*, siehe Abschlussbericht des Teilprojekts B2). In der dritten Förderungsperiode sollen diese Techniken erweitert werden, um die Planung einzubeziehen. Dadurch wird erreicht, dass Änderungen von Teilen des Plans oder von während der Projektdurchführung erstellten Produkten gezielt analysiert werden und Konsequenzen für den weiteren Projektverlauf festgestellt werden können. Dieses wiederum ermöglicht, den zuständigen Planer gezielt zu informieren, falls eine Planänderung notwendig wird, und gegebenenfalls Planungsaufgaben automatisch zu generieren (s.o.).

Softwareagenten zur Planungsunterstützung

Zur Unterstützung der Planung und Ausführung durch menschliche Agenten kann man Softwareagenten konzipieren und realisieren [6]. Je nach Aufgabenbereich können diese eigenständig Aufgaben bearbeiten oder Assistenzsysteme realisieren, die den menschlichen Agenten, vom Manager über den Planer bis hin zum Programmierer, bei seiner Arbeit unterstützen. Hierbei werden sie sowohl unbearbeitete als auch aufgearbeitete und ausgewertete Daten zur Verfügung stellen.

In Zusammenarbeit mit dem Teilprojekt A2 sollen sowohl die Möglichkeiten der Automatisierung von Planungs- und Ausführungsaufgaben als auch mögliche Konzeptionen von Assistenzsystemen untersucht werden, die den Planer wissensbasiert unterstützen. Dazu sollen Ansatzpunkte zur automatischen Unterstützung identifiziert, ausgearbeitet und mit einer Wissensmanagement-Komponente abgeglichen werden, wie sie z.B. im Teilprojekt A2 in der nächsten Förderungsperiode entstehen wird. Zusätzlich sollen in der dritten Förderungsperiode Möglichkeiten erforscht werden, dem Benutzer aktiv Hinweise zu geben, wenn sich die Möglichkeit einer Verbesserung ergibt, die im bisherigen Projektverlauf noch nicht berücksichtigt wurde. Ein

Beispiel für einen solchen Verbesserungsvorschlag ist der Hinweis für einen Projektmanager, dass jetzt ein Software-Agent mit einer besonderen Fähigkeit zur Verfügung steht, der in der Vergangenheit zwar einmal gesucht, aber nicht gefunden wurde.

Möglichkeiten der Automatisierung von Planungs- und Ausführungsschritten, etwa von Messwerterfassungen, Routineaufgaben bei der Projektdurchführung (etwa die Kompilierung von Source Code) und einzelnen Planungsaufgaben (z.B. Delegierung von Aufgaben, automatische Anpassung des Zeitplans, usw.) sollen im Laufe der dritten Förderungsperiode ebenfalls untersucht werden.

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Unterstützung der initialen Planung

Im Arbeitsschwerpunkt A sollen die Möglichkeiten zur Unterstützung der Vorplanung, d.h. der Erstellung eines initialen Planes vor Projektbeginn, untersucht werden. Dabei geht es zum einen um das Nachhalten der Planungsentscheidungen und deren Abhängigkeiten untereinander in einem Plan-Trace. Dieser kann dann verwendet werden, um Planungsaufgaben zu generieren und bei Änderungen den/die betroffenen Planer automatisch zu notifizieren. Zum anderen soll im Rahmen dieses Arbeitsschwerpunktes untersucht werden, wie die (lokale und verteilte) initiale Planung vom System wissensbasiert unterstützt und eventuell teilweise automatisiert werden kann.

Arbeitspaket A1: Identifikation von Möglichkeiten zur Planverbesserung

In diesem Arbeitspaket soll untersucht werden, wie aus dem momentanen Zustand des Plans die Notwendigkeit weiterer Planungsschritte abgeleitet werden kann. Ein Projektplan unterliegt gewissen Integritätsbedingungen, die am Ende des Prozesses der Metaplanung und vor Ausführung des eigentlich Planes erfüllt sein müssen, um dessen Konsistenz zu gewährleisten. Z. B. tritt ein inkonsistenter Zustand ein, wenn ein Input-Parameter einer Aufgabe von keiner anderen Aufgabe im Plan produziert wird oder wenn einer Aufgabe bisher noch keine Methode zur Vorgehensweise oder kein Bearbeiter zugewiesen ist. In diesem Fall kann eine möglichst spezifische Meta-Aufgabe automatisch generiert und in den Meta-Plan eingefügt werden.

Andere Situationen, in denen Meta-Aufgaben automatisch generiert werden sollten, können mit Hilfe von Meta-Modellen explizit spezifiziert werden. Beispielsweise ist es sinnvoll, für jede Implementierungs-Aufgabe im Plan auch eine Test-Aufgabe einzufügen. Durch die Generierung einer Meta-Aufgabe kann der zuständige Planer hieran erinnert werden, ohne ihm unnötig vorzugreifen, wie es bei der automatischen Einfügung dieser Aufgabe der Fall wäre. Im Rahmen von Arbeitspaket A1 sollen typische Planungssituationen sowohl in der Domäne der Gebäudeautomation als auch im Entwicklungsprozess des im Rahmen der Teilprojekte A2 und B2 implementierten MILOS-Systems untersucht und in Form von Meta-Modellen festgehalten werden. Außerdem muss dem Benutzer eine Möglichkeit zur Verfügung gestellt werden,

Meta-Modelle selbst zu erstellen oder existierende Default-Meta-Modelle an die Anforderungen eines speziellen Projekts oder einer speziellen Domäne anzupassen.
 Personalbedarf: 10 PM

Arbeitspaket A2: Generierung und Delegierung von Meta-Aufgaben

Aufbauend auf den Ergebnissen von Arbeitspaket A1 soll im zweiten Arbeitspaket des Arbeitsschwerpunkts A die Generierung von Meta-Aufgaben und deren Einfügung in den Meta-Plan verwirklicht werden. Hierzu muss untersucht werden, welche Art von Beziehungen zwischen den verschiedenen Meta-Aufgaben bestehen, so dass Abhängigkeiten innerhalb des Meta-Plans automatisch erfasst und verwaltet werden können. Solche Abhängigkeiten können aus einem Meta-Modell abgeleitet werden; beispielsweise kann eine Meta-Aufgabe zur Einfügung einer Test-Aufgabe in den Plan aus dem Meta-Plan entfernt werden, falls die zugehörige Implementierungs-Aufgabe aus dem Plan genommen wird. Andererseits können Abhängigkeiten auch durch allgemeine Regeln ausgedrückt werden. Z.B. besteht eine Abhängigkeit zwischen allen Planungsaufgaben, die die Zuordnung von Ressourcen mit ähnlichen Eigenschaften zu verschiedenen Aufgaben im Plan zum Ziel haben.

Nachdem eine Meta-Aufgabe generiert und in den Plan eingefügt wurde, muss sie einem geeigneten Planer zugeordnet werden und auf dessen To-Do-Liste erscheinen. In dem Fall, dass die Planung zentral durch einen einzelnen Planer stattfindet, ist diese Aufgabe trivial. Handelt es sich allerdings um eine verteilte Situation mit mehreren Planern, müssen Regeln erstellt werden, welchem Planer eine generierte Meta-Aufgabe zugeordnet werden soll. Eine solche Regel könnte beispielsweise lauten, dass alle Aufgaben zur Ressourcen-Zuordnung an den Verwalter des Resource-Pools delegiert werden sollen, oder dass alle Meta-Aufgaben, die sich auf einen bestimmten Teil des Gesamt-Plans beziehen, dem für diesen Teilplan zuständigen Planer zugeordnet werden müssen. Diese Regeln sind projektabhängig. Daher sollen im Rahmen dieses Arbeitspaketes sowohl anhand der im SFB vorhandenen Erfahrungen in der Entwicklung von Gebäude-Automationssystemen als auch anhand des Entwicklungs-Prozesses des MILOS-Systems sinnvolle Default-Regeln erstellt werden, die aber vom Benutzer für jedes Projekt ausgewählt und angepasst werden können.

Personalbedarf: 8 PM

Arbeitspaket A3: Entscheidungsabhängigkeiten und Plan Traces

Wie bereits oben erwähnt, sollen die Abhängigkeiten zwischen Meta-Aufgaben automatisch verwaltet und zur Generierung von Änderungsnotifikationen verwendet werden. Dazu ist es notwendig, den Trace des Planes, d.h. die Folge der vorgenommenen Planungsaktivitäten nachzuhalten, nämlich durch "Buchführung" der getroffenen Planungsentscheidungen und deren Abhängigkeiten untereinander. Zusätzlich zur automatischen Erfassung solcher Abhängigkeiten anhand der im Rahmen von Arbeitspaket A2 identifizierten Regeln soll hierzu dem Benutzer die Möglichkeit gegeben werden, weitere Abhängigkeiten zwischen getroffenen Planungsentscheidungen "von Hand" anzugeben.

In der zweiten Förderungsperiode wurde im Teilprojekt B2 eine Beschreibungssprache entwickelt, mit deren Hilfe Abhängigkeiten zwischen Planungs- und Ausführungsentscheidungen dargestellt und operationalisiert werden können (Design Rationales, siehe Abschlussbericht des Teilprojekts B2). Im Rahmen dieses Arbeitspaketes soll diese Sprache so erweitert werden, dass auch Abhängigkeiten zwischen komplexen Planungsentscheidungen dargestellt werden können. Um nachvollziehen zu können, in welcher Situation und mit welcher Begründung welche Planungsentscheidungen getroffen oder zurückgezogen wurden, muss der Plan-Trace außer den momentan gültigen Planungsentscheidungen (und deren Abhängigkeiten untereinander) auch die Historie der in der Vergangenheit getroffenen und zurückgezogenen Entscheidungen (und deren Zusammenhang mit anderen Entscheidungen) verwalten können.

Die oben erwähnte Sprache zur Erfassung von Design Rationales muss also in zweierlei Hinsicht erweitert werden:

- a) Die Sprache muss zusätzlich zu Abhängigkeiten zwischen Ausführungsentscheidungen auch die Darstellung von Zusammenhängen innerhalb der Planung zulassen.
- b) Zusätzlich zur Abhängigkeitsverwaltung zur Laufzeit müssen Konzepte entwickelt werden, die das Ablegen dieser Abhängigkeiten und der zeitlichen Zusammenhänge zwischen getroffenen Entscheidungen erlauben.

Personalbedarf: 7 PM

Arbeitspaket A4: Implementierung von Meta-Aufgaben und Plan-Traces in MILOS

Im Rahmen dieses Arbeitspaketes sollen die Ergebnisse der Arbeitspakete A1, A2 und A3 in dem in der letzten Förderungsperioden in den Teilprojekten A2 und B2 implementierten MILOS-System verwirklicht werden. Dazu soll die MILOS-Workflowengine so erweitert werden, dass sie außer der Durchführung von Aufgaben des Software-Entwicklungsprozesses auch die Ausführung von Meta-Aufgaben, also Aufgaben des Planungsprozesses, unterstützt. Insbesondere müssen dazu außer Produkten auch Plan-Teile als Parameter von Aufgaben zugelassen werden. Außerdem müssen die bisher in MILOS implementierten Abhängigkeiten so erweitert werden, dass auch Beziehungen zwischen Planteilen erfasst werden können. Soweit möglich sollen Meta-Aufgaben genauso behandelt werden, wie Aufgaben des Software-Entwicklungsprozesses: sie sollen auf der To-Do-Listen der zuständigen Person erscheinen, Input- und Output-Parameter haben und innerhalb des MILOS-Systems durchgeführt werden, indem das entsprechende Werkzeug automatisch gestartet und dem Planer ermöglicht wird, eine neue Version des Output-Produkts (in diesem Fall also eine neue Planversion) zu erstellen.

Um einen Plan-Trace halten zu können, muss die erweiterte MILOS-Workflowengine die im Arbeitspaket A3 entwickelten Konzepte zum Nachhalten von getroffenen und zurückgezogenen Planungsentscheidungen verwirklichen.

Für die Erstellung und Anpassung von Meta-Modellen und Regeln zur Generierung und Delegierung von Meta-Aufgaben werden dem Benutzer entsprechende Editoren zur Verfügung gestellt werden.

Personalbedarf: 18 PM

Arbeitspaket A5: Identifikation von Möglichkeiten zur wissensbasierten Planungsunterstützung und Automatisierung

Da es aufgrund der langen Projektlaufzeiten und schwer durchschaubaren Sachverhalte im Software-Engineering kaum möglich ist, alles relevante Wissen zu Beginn der Projektplanung schon zur Verfügung zu haben, sollte die Planung zusätzlich zu den in den ersten drei Arbeitspaketen des Arbeitsschwerpunkts A angesprochenen Möglichkeiten zur Planungsunterstützung durch Nachhalten von Entscheidung und Verwaltung von Abhängigkeiten zwischen diesen Entscheidungen auch aktiv durch wissensbasierte Hilfestellung unterstützt werden. In diesem Arbeitspaket sollen sowohl die Beispiel-Domäne des SFB, also die Domäne der Gebäudeautomation, als auch der Entwicklungsprozess des MILOS-Systems auf Chancen zur wissensbasierten Entscheidungshilfe hin untersucht werden. Die Resultate dieser Untersuchung sollen mit einer Wissensmanagement-Komponente abgeglichen werden, wie sie z.B. in der nächsten Förderungsperiode im Teilprojekt A2 entstehen soll. Diese Wissensmanagement-Komponente kann dann in den spezifizierten Situationen aktiv oder auf Nachfrage Informationen zur Planungsunterstützung liefern. Gebiete, in denen sich eine derartige Unterstützung anbietet, sind z.B.

- Anbieten von in einer bestimmten Planungssituation sinnvoll wiederverwendbaren Prozessmodellen bzw. alten (Teil-) Plänen
- Vorschläge zur Wiederverwendung von in ähnlichen Situationen früherer Projekte getroffenen Methoden-, Ressource-, Zeitplanungs- oder Messplanungsentscheidungen
- Vorschläge zur besten Vorgehensweise in einer bestimmten Planungssituation z.B. aufgrund von vorhandenen Qualitätsmodellen oder anderen Erfahrungen
- Unterstützung bei der Auswahl des am besten geeigneten Agenten für eine Aufgabe unter Berücksichtigung der Anforderungen der Aufgabe, der Fähigkeiten des Agenten und der Auslastung aller möglichen Agenten

Zusätzlich zu der Bereitstellung von Informationen zur Lösung von Aufgaben soll auch untersucht werden, in welchen Situationen Planungsentscheidungen aufgrund der vorhandenen Information automatisch getroffen werden können. Eine derartige Automatisierung wird in Form von Software-Agenten erfolgen, die im MILOS-System Planungsaufgaben übernehmen und selbständig ausführen.

Personalbedarf: 10 PM

Arbeitspaket A6: Dynamische Teambildung

Flexible Unternehmen bestehen nicht aus festen Teams, sondern sind in der Lage, diese projektspezifisch zusammenzustellen, abhängig von den Anforderungen, die

sich in dem speziellen Fall ergeben[5][7]. Die Teambildung ist also ein wichtiger, nicht trivialer Planungsschritt, der von einem Planungs- und Ausführungssystem unterstützt werden kann und sollte. Im Rahmen dieses Arbeitspakets sollen Regeln identifiziert werden, wie die Anforderungen an das zu bildende Team (anhand von Erfahrungen mit ähnlichen Projekten) aus den anstehenden Aufgaben eines neuen Projektes extrahiert werden können. Die Bildung des Teams aufgrund dieser Anforderungen kann dann vom System unterstützt werden, indem die Verhandlungen innerhalb des verteilten Unternehmens mit Hilfe von Softwareagenten durchgeführt werden, die einen globalen Überblick über die im Unternehmen vorhandenen Projekte und Ressourcen haben. Andererseits können Softwareagenten auch bei der Aufteilung des Projekts in Teilprojekte behilflich sein, indem sie Informationen zur Identifikation von sinnvollerweise ausgliederbaren Teilen eines Projektes zur Verfügung stellen oder sogar aktiv Vorschläge zu dieser Aufteilung machen.

Personalbedarf: 8 PM

Arbeitspaket A7: Implementierung von Softwareagenten zur Teambildung

Im Rahmen dieses Arbeitspaketes sollen die im Rahmen von Arbeitspaket A6 erarbeiteten Konzepte für Softwareagenten zur Unterstützung der Teambildung im MILOS-System implementiert werden. Das MILOS-System soll also zusätzlich zu den (menschlichen oder automatisierten) Planungs- und Ausführungsagenten auch Software-Agenten als Ressourcen zulassen, deren Aufgabe die Bereitstellung von Wissen zur Unterstützung der Teambildung ist.

Personalbedarf: 7 PM

Arbeitsschwerpunkt B: Unterstützung der Umplanung und Planverfeinerung

In diesem zweiten Arbeitsschwerpunkt sollen die Ergebnisse von Arbeitsschwerpunkt A so erweitert werden, dass außer der initialen Planung auch die Planverfeinerung und Umplanung nach Beginn der Plandurchführung durch Notifikationen und Bereitstellung von benötigten Informationen unterstützt wird. Hier liegt der Schwerpunkt der Betrachtung auf den Abhängigkeiten zwischen Planungsentscheidungen und solchen Entscheidungen, die während der Ausführung getroffen werden. Insbesondere muss auch der Plan-Trace so angepasst werden, dass Verbindungen zum Ausführungstrace darstellbar sind. Der Zustand der Planausführung kann wie der Plan-Zustand selbst automatisch analysiert werden, um von Systemseite aus Vorschläge und Hinweise zu möglichen Schritten zur Anpassung und Verbesserung des Plans zu geben.

Arbeitspaket B1: Untersuchung von Veränderungen des Ausführungszustands, die weitere Planschritte notwendig machen

Wie aus dem Planzustand, kann auch aus dem momentanen Zustand der Planausführung die Notwendigkeit weiterer (Um-)Planungsschritte abgeleitet werden. Z.B. muss der Plan angepasst werden, wenn sich während der Ausführung herausstellt, dass die Zeitplanung für eine Aufgabe oder einen Teilplan unrealistisch ist oder dass für eine Aufgabe andere Ressourcen oder Inputs gebraucht werden, als bisher angenommen.

Die Planung kann unter Umständen von Ergebnissen der Planausführung abhängen, z. B. kann die Implementierung nicht detailliert geplant werden, bevor nicht der Systementwurf festgelegt hat, welche Komponenten zu implementieren sind. Darüber hinaus ist es möglich, dass Meta-Aufgaben erst zur Ausführungszeit festgelegt werden können. So kann z.B. ein in der Erfahrungsdatenbank vorhandenes Qualitätsmodell festlegen, dass die Planung für eine Test-Aufgabe überdacht werden muss, wenn die zu testende Komponente eine gewisse Komplexität überschreitet.

In diesem Arbeitspaket sollen die beiden Beispiel-Projekte (Gebäudeautomatisierung und MILOS-Entwicklung) daraufhin untersucht werden, wie der Meta-Plan von dem Zustand der Planausführung abhängt. Das identifizierte Wissen soll in Form von Regeln zur Generierung und Verwaltung von Meta-Aufgaben festgehalten werden. Wie auch in Arbeitspaket A2 müssen außerdem Regeln zur Delegation der erzeugten Meta-Aufgaben festgelegt werden.

Personalbedarf: 6 PM

Arbeitspaket B2: Verzahnung von Plan- und Ausführungs-Traces

Da Planungs- und Ausführungsentscheidungen voneinander abhängen können, kann auch das Nachhalten dieser Aktivitäten nicht getrennt erfolgen, sondern muss verzahnt geschehen. Im Rahmen dieses Arbeitspaketes soll daher die in MILOS bereits existierende (und im Arbeitspaket A4 erweiterte) Sprache zur Abhängigkeitsverwaltung so angepasst werden, dass nicht nur Abhängigkeiten innerhalb von Planung und Ausführung verwaltet werden können, sondern auch Zusammenhänge zwischen diesen beiden Stufen von Aktivitäten darstellbar sind. Die in Arbeitspaket A3 entwickelten Konzepte zur Verwaltung der Historie von Planungsentscheidungen müssen so erweitert werden, dass auch die Historie der Planausführung und der während der Ausführung getroffenen Entscheidungen zur Planverfeinerung und Umplanung (einschließlich der Verzahnung dieser Entscheidungen mit während der Planausführung entstandenen Daten) mit ihrer Hilfe darstellbar sind.

Die so mitprotokollierten Abhängigkeiten sollen dann zur Versendung von Änderungsnotifikationen an Planer und Ausführer verwendet werden, wenn eine Änderung des Plan- oder Ausführungszustands weitere Änderungen (an Plan oder Ausführungszustand) notwendig macht.

Personalbedarf: 5 PM

Arbeitspaket B3: Implementierung in MILOS

In diesem Arbeitspaket sollen die in Arbeitspaket B1 identifizierten Regeln zur Generierung und Verwaltung von Meta-Aufgaben in die in Arbeitspaket A4 entwickelte Workflowengine zur Meta-Plan-Verwaltung integriert werden. Außerdem sollen die dort implementierten Konzepte zum Nachhalten von Plan-Traces um die in Arbeitspaket B2 identifizierten Konzepte zur Verzahnung von Plan- und Ausführungstrace erweitert werden.

Personalbedarf: 12 PM

Arbeitspaket B4: Wissensbasierte Unterstützung und Automatisierung der Umplanung und Planverfeinerung während der Planausführung

Hier sollen die in Arbeitspaket A5 erarbeiteten Konzepten zur wissensbasierten Unterstützung (und teilweisen Automatisierung) der initialen Planung so erweitert werden, dass sie auch Umplanungs- und Planverfeinerungsaktivitäten während der Plandurchführung einschließen. Z.B. könnte eine Wissensmanagement-Komponente dem Planer oder Ausführer die folgenden Informationen proaktiv oder auf Nachfrage zur Verfügung stellen:

- Vorschläge zur Plankorrektur bei Abweichungen der Ausführung von der Planung, z.B. aufgrund vorhandener Qualitätsmodelle
- Vorschläge zur Wiederverwendung von Produkten, die in vergangenen Projekten in ähnlichen Situationen entstanden sind (vor allem Wiederverwendung von Produkten, die in Zusammenhang mit einem in diesem Projekt wiederverwendeten Plan-Trace entstanden sind)

Sowohl die SFB-Beispieldomäne der Gebäudeautomation als auch der MILOS-Entwicklungsprozess sollen auf Möglichkeiten einer derartigen wissensbasierten Unterstützung der verzahnten Planung und Ausführung hin untersucht werden. Außerdem soll auch die Automatisierung von Umplanungsaufgaben betrachtet werden. Die Ergebnisse dieser Untersuchung sollen wie die Ergebnisse von Arbeitspaket A5 in eine Wissensmanagement-Komponente eingebracht werden, z.B. die im Rahmen des Teilprojekts A2 in der nächsten Förderungsperiode zu entwickelnde Wissensmanagement-Komponente.

Personalbedarf: 5 PM

Arbeitspaket B5: Verteilten Projektplanung und Ausführung in dynamischen Teams

In diesem Arbeitspaket sollen Möglichkeiten untersucht werden, den Mitgliedern räumlich verteilter Teams (z.B. dynamisch gebildeter Teams, siehe Arbeitspaket A6) bei der Planung und Durchführung von Softwareentwicklungs-Projekten Unterstützung an die Hand zu geben. Hierzu müssen sowohl die in der letzten Förderungsperiode im Rahmen des Teilprojekts B2 entwickelten Notifikationen während der Projektdurchführung als auch die in den Arbeitspaketen A3 und B2 dieses Antrags erarbeiteten Notifikationen zur Planungsunterstützung dahingehend klassifiziert werden, ob sie über Teamgrenzen hinweg versendet werden dürfen oder sollen, oder ob sie aus Gründen der Geheimhaltung oder zur Vermeidung von Informationsüberlastung nur innerhalb eines Teams sinnvoll sind. Ähnliches gilt für den Fluss von Daten und Arbeitsergebnissen zwischen den verschiedenen Teams.

In verteilten Teams noch stärker als bei lokaler Software-Entwicklung muss darauf geachtet werden, dass jedes einzelne Team seine Milestones und Deadlines einhält, vor allem solche, die sich über Teamgrenzen hinweg auswirken: In dynamisch gebildeten Teams (wie z.B. einzelnen Profit-Zentren eines Unternehmens), die ihre Zusammenarbeit anhand von verbindlichen Bedingungen "vertraglich" festgelegt haben, kann wegen des eingeschränkten Informationsflusses zwischen den Teams nur

schlecht auf Planabweichungen reagiert werden. Es bietet sich also an, den Teammitgliedern Softwareagenten zur Seite zu stellen, die aufgrund von vorhandenem Erfahrungswissen Ratschläge zur Vermeidung von Deadline-Verletzungen geben oder, wenn sich eine Verletzung nicht mehr vermeiden lässt, dem Benutzer bei der Aushandlung neuer Bedingungen zur Seite stehen.

Personalbedarf: 10 PM

Arbeitsschwerpunkt C: Praktischer Einsatz

In diesem Arbeitsschwerpunkt sollen die in den ersten beiden Schwerpunkten erarbeiteten Konzepte im praktischen Einsatz evaluiert werden, indem das MILOS-System, in dem diese Konzepte implementiert werden sollen, sowohl im Entwicklungsprozess des MILOS-Systems selbst zum Einsatz kommt, als auch in dem von Teilprojekt A1 verwalteten SFB-weiten SE-Labor installiert und zur allgemeinen Benutzung freigegeben wird.

Arbeitspaket C1: Einsatz im MILOS-Entwicklungsprozess

Um das MILOS-System im Software-Entwicklungsprozess der Teilprojekte A2 und B2 (die beide im Rahmen der nächsten Förderungsperiode Weiterentwicklungen an dem System vornehmen werden) einsetzen zu können, muss dieser Prozess in MILOS modelliert und geplant werden. Nach Fertigstellung der Planung kann das System dann zur Unterstützung der (zwischen der Universität Kaiserslautern und der University of Calgary verteilten) Software-Entwicklung an dem MILOS-Prototypen zum Einsatz kommen. Während dieses Einsatzes aufgedeckte Fehler und Verbesserungswünsche werden mitprotokolliert und behoben werden.

Personalbedarf: 22 PM

Arbeitspaket C2: Einbringung in den Gesamt-SFB

Vor einem möglichen Einsatz im SE-Labor muss das (auf Windows-NT entwickelte und getestete) MILOS-System auf heterogenen Plattformen installiert und eventuell angepasst werden. Sobald das System installiert und einsatzfähig ist, werden Mitarbeiter der Teilprojekte A2 und B2 den Mitgliedern des SFB beratend bei der Benutzung von MILOS zur Seite stehen. Die im Laufe des praktischen Einsatzes zu erwartenden Wartungsarbeiten werden in Zusammenarbeit zwischen den Teilprojekten A2 und B2 vorgenommen.

Personalbedarf: 16 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.		
Grundausrüstung Kötting, Boris								
	A6			A7	B5			
				A7	B5			
	C1			C2				
Ergänzungsausrüstung Holz, Harald								
	A3	B2		B3				
		A4		A5		B4		
	Goldmann, Sigrid		A1		A2		A4	
				B1		B3		
				C2				
	N.N., stud. Hilfskraft		C1		A7			
		A4			B3			
	N.N., stud. Hilfskraft		C1		C2			

Im Zeitplan ist ein Aufwand von wenigstens 8 Std. / Woche für SFB-übergreifende Themenstellungen berücksichtigt, die jedes Teilprojekt zur Verfügung stellt. Diese Arbeiten werden in Teams organisiert und im Arbeitsbericht gesondert beschrieben.

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Im Teilprojekt B1 werden Techniken und Methoden der Planung experimentell erprobt. Dies soll in der Implementierung des *Prozessleitstands*, eines Werkzeugs zur erfahrungsbasierten Unterstützung der Planung, resultieren, das im Rahmen des Teilprojekts A2 an das MILOS-System angebunden werden soll. Der Prozessleitstand wird dem Teilprojekt B2 wichtige Unterstützung bei der Generierung von Meta-Aufgaben zur Behebung von Planmissständen geben.

Die im Teilprojekt B2 in der nächsten Förderungsperiode zu identifizierenden Abhängigkeiten der Planung untereinander bzw. Maßnahmen zur Erhaltung der Plankonsistenz werden dem Teilprojekt A2 als Anforderungen für die pro-aktive wissensbasierte Assistenz für die Planung und Umplanung dienen. Die Ergebnisse des Teilprojektes

A2 sollen im MILOS-System implementiert werden und können somit von den im Teilprojekt B2 zu entwickelnden Agenten zur Planungsunterstützung verwendet werden.

Die in den D-Projekten untersuchte Domäne der Gebäudeautomatisierung wird als Grundlage für die Analyse der Anforderungen an ein in dieser Domäne einsetzbares Planungsunterstützungssystem dienen.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	2	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
			StHkr	2	40.800	StHkr	2	40.800	StHkr	2	40.800
	zus.:	2	zus.:	4	249.600	zus.:	4	249.600	zus.:	4	249.600
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		3.000	(522)		3.000	(522)		3.000
			zusammen		3.000	zusammen		3.000	zusammen		3.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausstattung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Richter, M. M., Prof. Dr. (3) Kötting, B., Dipl-Inform. (4) N.N., Dipl-Inform.	Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	5-10 20 20	1.1.95 1.1.98 1.1.01	
Ergänzungsausstattung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Holz, H., Dipl.-Inform. (2) Goldmann, S., Dipl.-Inform. (3) N.N., StHkr (4) N.N., StHkr	Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	38,5 38,5 19 19	1.9.96 1.2.98 1.1.01 1.1.01	BAT IIa BAT IIa StHkr StHkr
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Die wissenschaftlichen Mitarbeiter werden benötigt, um die geplanten Arbeiten im beschriebenen Umfang durchführen zu können. Aufgrund der Art der Aufgabenstellung handelt es sich um schwierige ingenieurmäßige Entwurfs-, Planungs- und Entwicklungstätigkeiten.

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Der erste Mitarbeiter der Grundausrüstung wird vornehmlich an Aufgaben der dynamischen Teambildung in großen Unternehmen arbeiten. Dabei wird ihm der zweite Mitarbeiter der Grundausrüstung zur Hand gehen. Außerdem wird der zweite Mitarbeiter der Grundausrüstung die Verantwortung für die Einführung und Evaluation des MILOS-Systems sowohl in der Systementwicklung von MILOS selbst als auch im Gesamt-SFB übernehmen.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Der erste Mitarbeiter der Ergänzungsausrüstung wird an der Dokumentation und Wiederverwendung von Plan- und Ausführungsstraces arbeiten. Außerdem wird dieser Mitarbeiter die Untersuchung von Möglichkeiten zur wissensbasierten Unterstützung der Planung und Umplanung übernehmen.

Die zweite Mitarbeiterin der Ergänzungsausrüstung wird ihren Fokus auf die situationsbezogene Generierung von Meta-Aufgaben legen. Außerdem wird sie sich an der Einführung und Evaluation von MILOS im Gesamt-SFB beteiligen.

Die beiden wissenschaftlichen Hilfskräfte werden sich an der Implementierung der in der dritten Förderungsperiode entwickelten Konzepten beteiligen, sowie an den durch den Einsatz von MILOS in der Entwicklung des MILOS-Systems selbst und im Gesamt-SFB anfallenden Wartungsarbeiten mitarbeiten.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausstattung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausstattung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausstattung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausstattung der Sächlichen Verwaltungsausgaben

Besonders im Bereich des Knowledge- und Workflow-Managements wächst die Zahl der kommerziellen Produkte rasant an. Da es weder finanziell noch zeitlich machbar ist, sich mit der vollen Marktbreite der Produkte auseinanderzusetzen, müssen Studien von Unternehmen, die sich auf die Evaluierung dieser Produkte spezialisiert haben, eingekauft werden. Außerdem werden die sächlichen Verwaltungskosten in der Ergänzungsausstattung für kleinere Geräteerweiterungen und elektronisches Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
-	-	-	-

3.1 Allgemeine Angaben zum Teilprojekt B4

3.1.1 Thema

Generische Kommunikationssysteme

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, Rechnernetze, Datenkommunikation, Kommunikationsprotokolle, Formale Beschreibungstechniken

3.1.3 Leiter

Prof. Dr. Reinhard Gotzhein

geb. 9.10.1956

Universität Kaiserslautern, Postfach 3049, 67653 Kaiserslautern

Tel.: 0631/205-3426

Fax: 0631/205-3956

Email: gotzhein@informatik.uni-kl.de

URL: <http://rn.informatik.uni-kl.de/~gotzhein>

Die Stelle des Leiters ist befristet ☒ ja ☐ nein, bis zum

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ ja	☒ nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ ja	☒ nein
Tierversuche	☐ ja	☒ nein
gentechnologische Untersuchungen	☐ ja	☒ nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	625,2	6,0	0	631,2
1998	201,6	3,0	0	204,6
1999	206,4	3,0	0	209,4
2000	208,8	3,0	0	211,8
Zwischensumme	1242,0	15,0	0	1257,0
2001	249,6	3,0	0	252,6
2002	249,6	3,0	0	252,6
2003	249,6	3,0	0	252,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Große Systeme sind heute in der Regel nebenläufig und verteilt. Aus diesem Grund sind Kommunikationssysteme meist ein integraler Bestandteil großer Systeme, sie bilden die Basis für Betriebssysteme und Anwendungen. Die Anforderungen an Kommunikationssysteme sind bereits heute - bedingt durch verschiedenartige Anwendungen und Technologien - sehr vielfältig. Aus diesem Grund können sie nicht mehr durch wenige typische, möglichst viele Erfordernisse abdeckende Protokollstapel befriedigt werden. Zukünftige Kommunikationssysteme - ähnlich wie auch Betriebssystemplattformen - werden daher in zunehmendem Maße anforderungsangepasst sein. Dies verlangt u.a. andere Protokollarchitekturen, zusätzliche Protokollfunktionalitäten sowie schlankere und stärker integrierte Kommunikationsprotokolle. Ein entsprechender Entwicklungstrend zeichnet sich seit einigen Jahren ab. Dabei werden Anforderungen an die Dienstgüte (Quality of Service) von Kommunikationssystemen verstärkt zu berücksichtigen sein.

Langfristiges Ziel dieses Teilprojekts ist die Bereitstellung und Anwendung von Verfahren und Techniken zur Entwicklung anwendungsangepasster, maßgeschneiderter Kommunikationssysteme unter besonderer Berücksichtigung von Echtzeitanforderungen. Dabei wird der Vielfalt der Kommunikationsanforderungen sowie der Variabilität der Basistechnologien durch die generische Auslegung von Kommunikationssystemen und Entwicklungsprozessen Rechnung getragen. Die Qualität maßgeschneiderter Kommunikationssysteme wird durch die Bereitstellung domänenspezifischer Methoden, formaler Techniken und rechnergestützter Werkzeuge sichergestellt. Speziell entwickelte und international genormte formale Beschreibungstechniken (FDTs) spielen bereits seit mehr als 15 Jahren eine wichtige Rolle bei der Entwicklung von Kommunikationssystemen.

Das langfristige Ziel des Teilprojekts soll durch die folgenden, sich ergänzenden Forschungsschwerpunkte erreicht werden:

- a) Bereitstellung von formalen Methoden, Techniken und Werkzeugen zur generischen Auslegung von Kommunikationssystemen unter besonderer Berücksichtigung von Echtzeitanforderungen
- b) Erfassung, Verbesserung und Wiederverwendung der relevanten Aspekte des Entwicklungsprozesses von Kommunikationssystemen
- c) Erprobung der Methoden, Techniken und Werkzeuge zur Entwicklung generischer Kommunikationssysteme im Anwendungsfeld "Gebäudeautomation"

Darüberhinaus soll das Teilprojekt Beiträge zu den anderen Projektbereichen liefern.

3.3 Stand der Forschung

Die nachfolgend beschriebenen Arbeitsschwerpunkte B und C bilden eine thematische Verbreiterung des Teilprojekts gegenüber den Förderperioden 1 und 2. Aus diesem Grund wird hier der Stand der Forschung für diese Arbeitsschwerpunkte dargestellt.

Komponentenbasierte Protokollentwicklung

Wiederverwendungsansätze werden im Software Engineering seit geraumer Zeit untersucht. Ganz grob lässt sich die in Abbildung 1 dargestellte Klassifikation vornehmen [6]:

- *Komponenten* sind in sich abgeschlossene, sofort benutzbare (Kode-)Bausteine, die aus Bibliotheken selektiert und komponiert werden. Bekannte Beispiele sind objektorientierte Klassenbibliotheken, z.B. Java Packages, und Sun's Enterprise Java Beans.
- *Frameworks* sind vom Anwendungsentwickler zu ergänzende Applikations-skelette. Sie sind abstrakter als Komponenten und repräsentieren eine Menge von Entwurfsentscheidungen.
- *Muster* beschreiben generische Lösungen für wiederkehrende Entwicklungsprobleme, die aus einer Sammlung selektiert, an einen speziellen Kontext angepasst und komponiert werden. Sie sind abstrakter und feingranularer als Frameworks. Ein bekanntes Beispiel sind die Design Patterns [4].

Zu jedem dieser Ansätze liegen zahlreiche Ergebnisse und Anwendungen vor, wobei Frameworks häufig proprietären Charakter haben. Im Rahmen dieses Teilprojekts wurden zwei neue Ansätze entwickelt bzw. mitentwickelt, der *SDL-Pattern-Ansatz* und der *Requirement-Pattern-Ansatz*. Zusätzliches Wiederverwendungspotenzial liegt in komponentenbasierten Ansätzen, die hier zunächst kurz aus der allgemeineren

Sicht des Software Engineering und anschließend aus der spezielleren Sicht des Protocol Engineering betrachtet werden sollen.

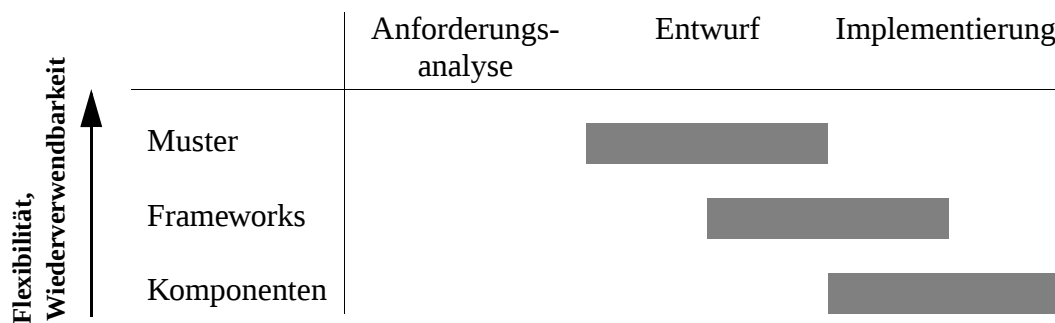


Abbildung 1: Klassifikation von Wiederverwendungskonzepten

In den letzten Jahren hat das industrielle und akademische Interesse an Softwarekomponenten und komponentenbasiertem Software Engineering massiv zugenommen. Wenngleich eine ganze Reihe von Definitionen des Begriffes im Umlauf sind und eine exakte Abgrenzung fehlt, so scheint doch Konsens in folgenden Punkte zu bestehen [2]:

- Es existiert eine untrennbare Beziehung zwischen Komponenten und Architektur. Die Architektur eines Systems definiert Anforderungen an die Art der Komponenten, insbesondere an die Beschaffenheit der Komponentenschnittstellen. Prominente Vertreter sind die Common Object Request Broker Architecture (CORBA) und das Distributed Component Object Model (DCOM).
- Es besteht kein unmittelbarer Zusammenhang zwischen Komponenten- und Objekttechnologie. Komponentenbasierte Entwicklung ist also weder auf objektorientierte Ansätze reduzierbar noch setzt sie solche Ansätze voraus.

Komponentenbasierte Ansätze setzen geeignete Modularisierungen mit entsprechender sprachlicher Unterstützung voraus. Im Protokollbereich lassen sich mehrere Architekturen, die bestimmte Modularisierungen implizieren, identifizieren:

- Klassischer Vertreter ist die *protokollbasierte (Schichten-)Architektur*, die den bekannten Protokollstapeln (TCP/IP, PROFIBUS, MAP) zugrundeliegt. Komponenten sind in diesen Fällen Protokolle bzw. Protokollinstanzen.
- Ein alternativer Vertreter ist die *funktionsbasierte Architektur* [18], mit Komponenten, die sich an Protokollfunktionalitäten orientieren.

Die protokollbasierte Architektur dient primär der Strukturierung der Kommunikationsaufgabe durch eine Modularisierung, die den Austausch von Komponenten durch funktional gleichwertige ermöglichen soll. Gleichwohl ist etwa bei TCP/IP die Vermengung der Funktionalitäten so groß, dass keine sauberen Schnittstellen vorhanden sind. Die Komponenten eines Protokollstapels sind in der Regel sehr grobgranular, was ihre Flexibilität stark einschränkt. Ferner sind die Komponenten oft unzureichend aufeinander abgestimmt, was sich in fehlenden bzw. mehrfach vorhandenen Funktio-

nalitäten äußert. Daher sind sie zur Konfigurierung von Kommunikationssystemen nur sehr bedingt geeignet.

Die funktionsbasierte Architektur dient ebenfalls der Strukturierung der Kommunikationsaufgabe; darüberhinaus soll sie die Konfigurierung von Protokollen unterstützen, so dass der Wiederverwendungsaspekt hier stärker im Vordergrund steht. Protokollfunktionalitäten sind in der Regel sehr feingranular. Eine besondere Schwierigkeit ist die Definition in sich abgeschlossener Funktionalitäten, so dass bei der Selektion und Komposition derartiger Komponenten starke Abhängigkeiten bestehen.

Im folgenden sind einige der bekannteren Ansätze zur Protokollkonfigurierung aufgeführt:

- *F-CSS (Function-based Communication Subsystem, [18, 19])* ist ein an der Universität Karlsruhe entwickelter Ansatz. Er unterstützt die dynamische Konfigurierung auf Basis einer funktionsbasierten Architektur. Dienstnutzer fordern Sitzungen an, die aus mehreren separaten Verbindungen bestehen können. Für jede Verbindung wird eine separate Protokollimplementierung konfiguriert bzw. eine vorkonfigurierte Implementierung verwendet. Zur Synchronisation der Verbindungen existiert ein spezieller Sitzungsmanager.
- *DaCapo (Dynamic Configuration of Protocols, [13, 14])*, entwickelt an der ETH Zürich, verwendet ebenfalls eine funktionsbasierte Architektur. Im Unterschied zu F-CSS wird jedoch nicht zwischen Verbindungen und Sitzungen unterschieden, so dass ein Protokoll bei Bedarf mehrere Datenströme verwaltet.
- Der *x-Kernel [9]* ist ein experimentelles Betriebssystem, das Anfang der 90er Jahre an der University of Arizona entwickelt wurde. Teil des x-Kernels ist eine Menge konfigurierbarer Protokollstapel, bestehend aus den klassischen Internet-Protokollimplementierungen, so dass hier eine protokollbasierte Architektur vorliegt. *AVOCA [11]* ist eine Weiterentwicklung des x-Kernels, bei der ähnlich wie bei F-CSS eine funktionsbasierte Architektur verwendet wird. Die Integration beider Vorgehensweisen findet in [1] statt.
- *ADAPTIVE (A Dynamically Assembled Protocol Transformation, Integration, and eValuation Environment, [15])* entstand an der University of Irvine. Architekturell besteht eine enge Verwandtschaft zu F-CSS, jedoch wird keine dynamische Konfigurierung angestrebt. Komponenten werden mit C++ unter Verwendung objektorientierter Techniken implementiert.
- *DyCAT (Dynamically Configurable and Adaptive Transport System, [7])*, entwickelt an der Universität Aachen, weist starke Parallelen zu F-CSS auf. Ziel ist auch hier die dynamische Konfigurierung von Protokollimplementierungen, wobei eine funktionsbasierte Architektur zugrundeliegt. Dabei kommen zur Modellierung Produktnetze zum Einsatz.

In allen genannten Fällen handelt es sich bei den Komponenten um *Kodebausteine*, die aus einer Bibliothek selektiert und anschließend komponiert werden, teilweise mit einem vorgegebenen Framework. Dies hat den Vorteil, dass das konfigurierte Proto-

koll unmittelbar übersetzt und ausgeführt werden kann, wie es beispielsweise bei dynamischer Konfigurierung notwendig ist. Nachteilig ist, dass Entwurfsentscheidungen nicht ohne weiteres revidiert werden können, da der Protokollentwurf nicht mitbetrachtet wird.

Automatische Generierung effizienter Protokollimplementierungen

Der Einsatz formaler Beschreibungstechniken im Protokollentwurf gewinnt mit der Möglichkeit zur automatischen Generierung von Protokollimplementierungen an Attraktivität. Der kostenintensive und fehleranfällige Schritt der manuellen Kodierung kann auf diese Weise vereinfacht oder sogar ganz eingespart werden. Kommerzielle Werkzeugumgebungen beinhalten daher Kodegeneratoren, die formale Entwurfsspezifikationen in Programme übersetzen, für die dann mit herkömmlichen Compilern ausführbarer Maschinencode erzeugt wird. Als Beispiele seien die Kodegeneratoren *CAdvanced* [16] des SDL Development Toolsets (SDT) und *ec* [3] des Estelle Development Toolsets (EDT) genannt.

Bei der automatischen Generierung von Implementierungen direkt aus formalen Spezifikationen treten jedoch gerade im Vergleich zu manuell erstellten Implementierungen vielfältige Probleme auf, die zum Teil auf Defizite der existierenden Kodegeneratoren zurückzuführen sind, zum Teil aber auch direkte Folgen des hohen Abstraktionsniveaus der formalen Beschreibungstechniken sind. Hier sind folgende Punkte besonders zu nennen [17]:

- Durch das hohe Abstraktionsniveau fehlen in der formalen Beschreibung viele zur Implementierung wichtige Details (z.B. Binärrepräsentationen von Datenwerten, Prüfsummenberechnungen).
- Eine Spezialisierung dieser Problematik bildet die Einbettung der aus formal spezifizierten Systemen automatisch generierten Implementierungen in existierende, reale Umgebungen mit ihren spezifischen Schnittstellen (z.B. Betriebssysteme, Netzwerke).
- Das hohe Abstraktionsniveau und die beschränkte Ausdrucksfähigkeit haben häufig eine Überspezifikation sowie eine für eine effiziente Implementierung ungeeignete Strukturierung zur Folge (z.B. nachrichtenbasierte Interaktion, Server-Architektur).

Zur Verringerung der Defizite existierender Kodegeneratoren existieren eine Reihe von Ansätzen. Ein Ansatz besteht darin, die Verwendung "teurer" Sprachkonstrukte in der Spezifikation zu verbieten, was einer Reduzierung der Ausdrucksfähigkeit entspricht. Dieser Ansatz wird mit dem Kodegenerator *CMicro* [12] des SDL Development Toolsets, der im Bereich der eingebetteten Systeme vermarktet wird, verfolgt. Alternativ kommt die Empfehlung bestimmter Spezifikationsstile in Betracht, wie sie z.B. in [5] (Nebenläufigkeit, strukturelle Transformationen) und [8] (Datenreferenzierung, Integrated Packet Framing) vorgeschlagen werden, was letzten Endes ebenfalls auf eine Reduzierung der Ausdrucksmöglichkeiten hinausläuft.

Ein weiterer Ansatz besteht in der Auswahl eines effizienzsteigernden Implementierungsmodells [8]. Kommerzielle Compiler übernehmen das jeweilige Modell der Spezifikationssprache, im Falle von SDL also das *Server-Modell*: jede Prozessinstanz wird als Server gesehen, der mit anderen Prozessinstanzen über asynchronen Signalaustausch interagiert. Dies erzeugt einen vergleichsweise hohen Kommunikationsaufwand, der durch das Hin- und Herkopieren von Signalen zu signifikanten Effizienzeinbußen führen kann. Hinzu kommt der Aufwand für das Scheduling der Prozesse und die Auswahl schaltbarer Transitionen. Diese Nachteile vermeidet das *Activity-Thread-Modell*: ein Activity-Thread wird durch das Eintreffen eines Signals kreiert und beinhaltet alle Verarbeitungsschritte, die sich daraus ergeben. Da dieses Modell synchrone Interaktionen voraussetzt, entspricht es offensichtlich nicht dem Modell von SDL. Damit es trotzdem als Implementierungsmodell eingesetzt werden kann, sind bestimmte Annahmen an die Struktur der Spezifikation zu treffen [17], die allerdings die Verwendbarkeit des Activity-Thread-Modells stark einschränken. Über Werkzeugunterstützung wird in [10] berichtet.

Literatur

- [1] Bhatti, N., Schlichting, R.: *A System for Constructing Configurable High-Level Protocols*, Proceedings of the ACM SIGCOMM'95 Symposium, 1995, S. 138-150
- [2] Brown, A. W., Wallnau, K. C.: *The Current State of CBSE*, IEEE Software, Vol. 15, No. 5, Sept./Okt. 1998, S. 37-46
- [3] Budkowski, S.: *Estelle Development Toolset*, Computer Networks and ISDN Systems, Vol. 25, No. 1, 1992
- [4] Gamma, E., Helm, R., Jonsson, R., Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, Massachusetts, 1995
- [5] Gotzhein, R., Brederke, J., Effelsberg, W., Fischer, S., Held, T., König, H.: *Improving the Efficiency of Automated Protocol Implementation Using Estelle*, Computer Communications 19 (1996), S. 1226-1235
- [6] Gotzhein, R., Schaible, P.: *Pattern-Based Development of Communication Systems*, in: Annals of Telecommunications, Special Issue on Protocol Engineering, Vol. 54, No. 11-12, 1999, S. 508-525
- [7] Heinrichs, B.: *DyCAT - A Flexible Transport System Architecture*, IEEE International Conference on Communications (ICC'93), Genf, Schweiz, 1993, S. 1331-1335
- [8] Henke, R., König, H., Mitschele-Thiel, A.: *Derivation of Efficient Implementations from SDL Specifications Employing Data Referencing, Integrated Packet Framing and Activity Threads*, in: SDL'97: Time for Testing - SDL, MSC

- and Trends, A. Cavalli and A. Sarma (Hrsg.), Elsevier Science, 1997, S. 397-414
- [9] Hutchinson, N., Peterson, L.: *The x-Kernel: An Architecture for Implementing Network Protocols*, IEEE Transactions on Software Engineering, Vol. 17, No. 1, 1991, S. 64-76
 - [10] Mitschele-Thiel, A., Langendörfer, P., Henke, R.: *Design and Optimization of High-Performance Protocols with the DO-IT Toolbox*, in: J. Bredereke, R. Gotzhein (Hrsg.), Protocol Specification, Testing and Verification XVI / Formal Description Techniques IX, Chapman & Hall, 1996, S. 45-60
 - [11] O'Malley, S., Peterson, L.: *A Dynamic Network Architecture*, ACM Transactions on Computer Systems, Vol. 10, No. 2, 1992, S. 110-143
 - [12] O'Donnell, R., Waldt, B., Bergstrand, J.: *Automatic Code for Embedded Systems Based on Formal Methods*, Technical Report, Telelogic AB
 - [13] Plagemann, T., et. al.: *A Model for Dynamic Configuration of Light-Weight Protocols*, in: IEEE Third Workshop on Future Trends of Distributed Computing Systems, Taipei, Taiwan, April 1992, S. 100-106
 - [14] Plagemann, T., et. al.: *Modules as Building Blocks for Protocol Configuration*, in: International Conference on Network Protocols, IEEE Computer Society Press, USA, 1993
 - [15] Schmidt, D. C., Box, D. F., Suda, T.: *ADAPTIVE: A Dynamically Assembled Protocol Transformation, Integration, and eValuation Environment*, Concurrency Practice and Experience, Vol. 5, No. 4, 1993
 - [16] Telelogic Tau 3.5 User's Manual, Part 6: *SDT Code Generators*, 1999
 - [17] Thees, J.: *Implementierungs- und Optimierungsmodelle des Experimental Estelle Compilers*, in: K. Spies, B. Schätz (Hrsg.): Formale Beschreibungstechniken für verteilte Systeme (FBT'99), Herbert Utz Verlag Wissenschaft, München, 1999
 - [18] Zitterbart, M., Stiller, B., Tantawy, A. N.: *A Model for Flexible High-Performance Subsystems*, IEEE Journal on Selected Areas in Communications, Vol. 11, No. 4, Mai 1993, S. 507-518
 - [19] Zitterbart, M.: *Flexible und effiziente Kommunikationssysteme für Hochleistungsnetze*, International Thomson Publishing, 1995

3.4 Eigene Vorarbeiten

Die in den Arbeitsschwerpunkten A und D geplanten Arbeiten der 3. Förderperiode setzen auf den bislang erzielten Ergebnissen des Teilprojekts B4 auf. Diese Ergebnisse sind in den Arbeits- und Ergebnisberichten der Förderperioden 1 und 2 ausführlich dargestellt.

Ein wesentliches Ergebnis der Förderperioden 1 und 2 sind die Entwicklung und der Einsatz musterbasierter Vorgehensweisen zur Ermittlung von Anwendungs- und Kommunikationsanforderungen sowie zur Maßschneiderung von Protokollen. In der Anforderungsphase kommen *Requirement Patterns* zur Anwendung, die aus einer bestehenden Mustersammlung selektiert, an die spezielle Problemstellung adaptiert und schließlich komponiert werden [25] [28]. In der Protokollentwurfsphase werden spezialisierte Protokollmuster, die von Teilprojekt B4 unter der Bezeichnung *SDL-Patterns* in die Literatur eingeführt worden sind, eingesetzt [22] [24].

Die musterbasierten Ansätze wurden in mehreren umfangreichen Fallstudien erprobt [23] [27] [29]. Die dabei gesammelten Erfahrungen sind jeweils unmittelbar in die Weiterentwicklung und Konsolidierung dieser Ansätze eingeflossen [20] [21]. Erste Arbeiten zur Integration der musterbasierten Ansätze sind in [26] dargestellt.

Literatur

- [20] Feldmann, R.L, Geppert, B., Rößler, F.: *Continuous Improvement of Reuse-Driven SDL System Development*, Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering (SEKE'99), Kaiserslautern, Juni 1999
- [21] Feldmann, R.L, Geppert, B., Rößler, F.: *An Integrating Approach for Developing Distributed Software Systems - Combining Formal Methods, Software Reuse, and the Experience Base*, Proceedings of the 5th International Conference on Engineering of Complex Computer Systems (ICECCS'99), Las Vegas, USA, Oktober 1999
- [22] Geppert, B., Gotzhein, R., Rößler, F.: *Configuring Communication Protocols Using SDL Patterns*, in: A. Cavalli, A. Sarma (Eds.), *SDL'97 - Time for Testing*, Elsevier Science Publishers, Proceedings of the 8th SDL-Forum, Evry, France, September 1997
- [23] Geppert, B., Kühlmeyer, A., Rößler, F., Schneider, M.: *SDL-Pattern based Development of a Communication Subsystem for CAN*, Proceedings of the IFIP Joint International Conference FORTE/PSTV'98, Paris, Frankreich, Kluwer Academic Publishers, November 1998
- [24] Geppert, B., Rößler, F.: *Generic Engineering of Communication Protocols - Current Experience and Future Issues*. Proceedings of the 1st IEEE International Conference on Formal Engineering Methods (ICFEM'97), Hiroshima, Japan, November 1997

- [25] Gotzhein, R., Kronenburg, M., Peper, C.: *Reuse in Requirements Engineering: Discovery and Application of a Real-Time Requirement Pattern*, 5th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'98), Lecture Notes in Computer Science 1486, Springer, Lyngby, Denmark, September 14-15, 1998. pp. 65-74
- [26] Gotzhein, R., Schaible, P.: *Pattern-Based Development of Communication Subsystems*, in: Annals of Telecommunications, Special Issue on Protocol Engineering, Vol. 54, No. 11-12, 1999, pp. 508-525
- [27] Kronenburg, M., Peper, C.: *Application of the FOREST Approach to the Light Control Case Study*, Journal of Universal Computer Science (J.UCS), Special Issue on "Requirements Engineering: The Light Control Case Study", Springer, 2000 (accepted for publication)
- [28] Peper, C., Gotzhein, R., Kronenburg, M.: *A Generic Approach to the Formal Specification of Requirements*. Proceedings of the 1st IEEE International Conference on Formal Engineering Methods (ICFEM'97), Hiroshima, Japan, November 1997, pp. 252-261
- [29] Rößler, F., Geppert, B., P. Schaible, P.: *Re-Engineering of the Internet Stream Protocol ST2+ with Formalized Design Patterns*, Proceedings of the 5th International Conference on Software Reuse (ICSR5), Victoria, British Columbia, Canada, 1998

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Ziel des Teilprojekts in der nächsten Förderperiode ist die Verbesserung, Verbreiterung, Integration und praktische Erprobung von Methoden, Techniken und Werkzeugen zur generischen Entwicklung maßgeschneiderter Kommunikationssysteme.

Ausgangspunkt für die geplanten Arbeiten sind die Ergebnisse der Förderperioden 1 und 2, insbesondere die musterbasierten Ansätze zur Anforderungsdefinition und Protokollentwicklung sowie die vorhandenen Mustersammlungen. Praktisches Ziel ist das *Engineering* realer, maßgeschneiderter und formal spezifizierter Kommunikationsmiddleware für den SFB-Anwendungsbereich "Gebäudeautomation" durch die Anwendung einer verbreiterten Palette von Wiederverwendungsansätzen. Mit den nachfolgend kurz beschriebenen Arbeitsschwerpunkten sollen in diesem Gesamtspektrum besondere Akzente gesetzt werden, die zum einen der Verbesserung bereits erzielter Ergebnisse dienen, zum anderen das Spektrum der Methoden und Techniken verbreitern. Ein besonderes Anliegen ist dabei die Abstimmung und Integration der Ergebnisse.

Das Aufgabenspektrum bei der Entwicklung von Kommunikationssystemen ist in Abbildung 2 qualitativ dargestellt. Vertikal wird eine Differenzierung nach Entwicklungsschritten vorgenommen, horizontal eine Unterscheidung nach Wiederverwendungsansätzen. Das Diagramm wird gefüllt durch eine Auswahl der in Teilprojekt B4 relevanten Themen. Die umrahmten Bereiche des Diagramms ordnen die betreffenden Themen den Mitarbeitern von B4 zu und motivieren die Strukturierung des Teilprojekts in folgende Arbeitsschwerpunkte:

A. Musterbasierter Protokollentwurf

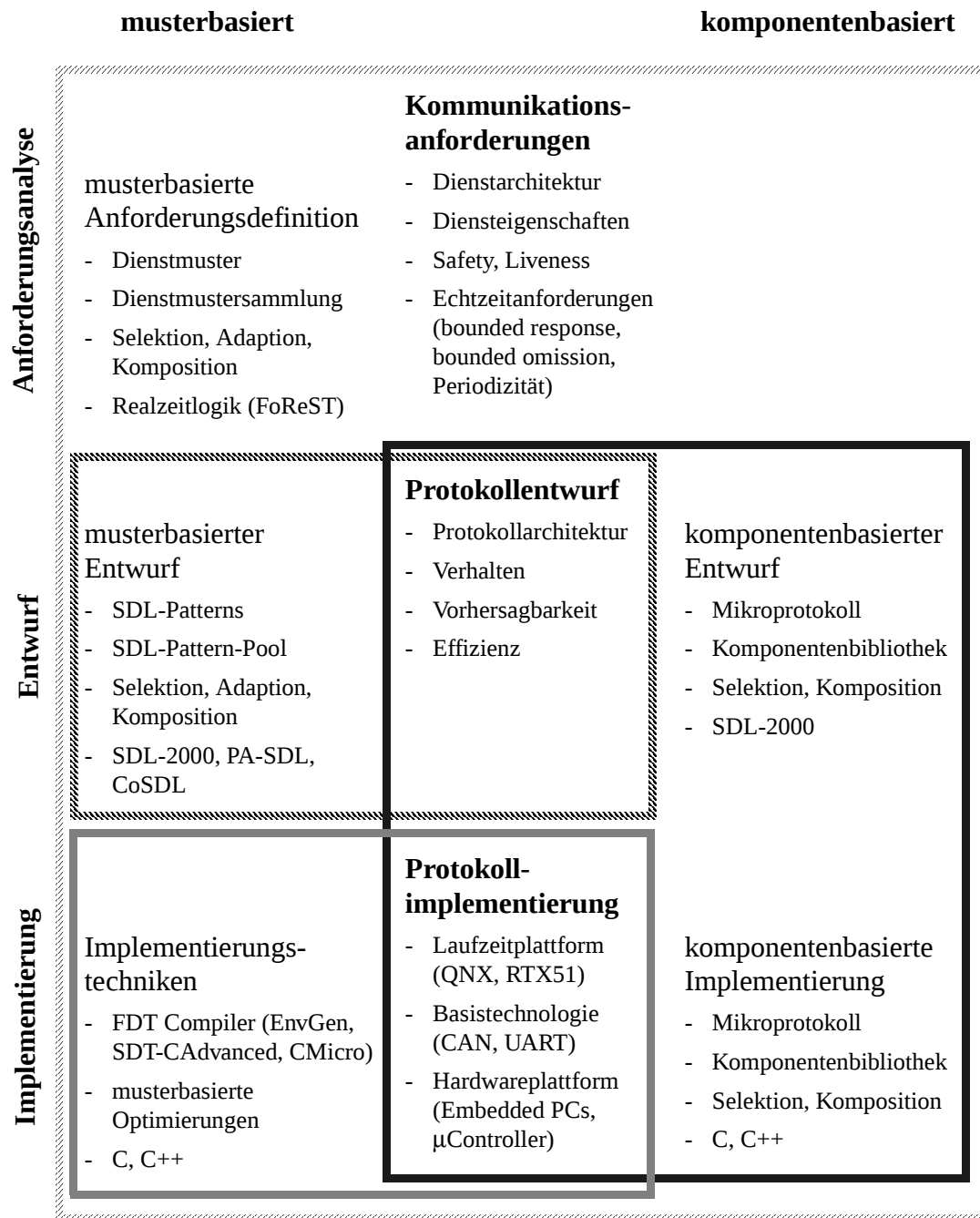
Ein wesentliches Ergebnis der vorangegangenen Förderperioden sind die Entwicklung und die praktische Erprobung des *SDL-Pattern-Ansatzes* für den Entwurf maßgeschneiderter Kommunikationsprotokolle. Bestandteile des Ansatzes sind eine Sammlung wiederverwendbarer Protokollbausteine in Form von SDL-Patterns sowie das inkrementelle Prozessmodell zur Erfassung der Entwicklungsmethodik. Zur Formulierung von Protokollmustern, die von B4 unter der Bezeichnung *SDL-Patterns* in die Literatur eingeführt worden sind, bzw. zu deren Instanziierung wird u.a. SDL (*Specification and Description Language*), eine international genormte, objektorientierte, graphische FDT (*Formal Description Technique*), verwendet (daher *SDL-Patterns*).

Ausgehend von den bisherigen Ergebnissen und Erfahrungen soll der SDL-Pattern-Ansatz konsolidiert und verbessert werden. Teil der Konsolidierung ist die Präzisierung der Beschreibung von SDL-Patterns und ihrer Anwendung. Zur Verbesserung tragen neue Analysemodelle sowie die Verbreiterung der bestehenden Mustersammlung bei.

B. Micro-Protocol Engineering

Um einen optimalen Wiederverwendungsgrad zu erzielen, ist es sinnvoll, mehrere Wiederverwendungsansätze zu kombinieren. Daher soll - zusätzlich zu der bereits untersuchten musterbasierten Vorgehensweise - der Komponentenansatz in die Protokollentwicklung einbezogen werden. Hier stellt sich nun zunächst die Frage, welche Arten von Komponenten sich für Zwecke des Protocol Engineering eignen. Protokoll- und funktionsbasierte Architekturen scheiden aus den bereits erörterten Gründen aus. Stattdessen soll ein neuartiger Ansatz, nämlich die Verteilung der Kommunikationsaufgabe auf eine Menge sog. *Mikroprotokolle*, untersucht werden.

Mikroprotokolle werden hier in erster Näherung erklärt als in sich abgeschlossene Protokolle mit sehr kleinem Funktionsumfang. Anders als Protokollfunktionalitäten sind Mikroprotokolle so konzipiert, dass die Wechselwirkung zwischen Komponenten sehr gering ist, was der Modularisierbarkeit und der Kompositionsfähigkeit entgegenkommt. Ihre Granularität bewegt sich zwischen Protokollstapelelementen und Protokollfunktionalitäten. Denkbar sind Mikroprotokolle für den Nutzdatenaustausch, die Systemüberwachung, die Rekonfigurierung, die dynamische Adressvergabe, die Adressauflösung und die Systeminitialisierung, wobei je nach Fehlermodell, Res-

Legende:

Arbeitsschwerpunkt A



Arbeitsschwerpunkt C



Arbeitsschwerpunkt B



Arbeitsschwerpunkt D

Abbildung 2: Aufgabenspektrum und Zuordnung zu Arbeitsschwerpunkten

sourcensituation und erforderlicher Dienstgüte unterschiedliche Mikroprotokolle komponiert werden könnten.

C. Ableitung effizienter Protokollimplementierungen

Die bisher eingesetzten Generierungstechniken dienen der Entwicklung von Prototypimplementierungen für eine QNX-Laufzeitplattform und haben die *Durchgängigkeit* des Entwicklungsansatzes belegt. Reale Implementierungen unterliegen jedoch sehr weitgehenden Einschränkungen in Bezug auf Hardware- und Laufzeitplattform, was grundsätzlich andere Implementierungsansätze erforderlich macht. Einige dieser Ansätze sollen im Kontext der Anwendungsdomäne auf ihre Eignung zur Implementierung maßgeschneiderter Kommunikationsmiddleware hin untersucht und weiterentwickelt werden.

D. Syntaktische und semantische Integration der Entwicklungsphasen

Ein wesentliches Ergebnis der Förderperioden 1 und 2 sind die Entwicklung und die Anwendung von musterbasierten Vorgehensweisen zur Ermittlung von Anwendungs- und Kommunikationsanforderungen sowie zur Maßschneiderung von Kommunikationsprotokollen. Diese Vorgehensweisen sind grundsätzlich in sich abgeschlossen, können also unabhängig voneinander angewendet werden. Allerdings wird bei einer isolierten Anwendung eines musterbasierten Ansatzes nur ein Teil des Wiederverwendungspotenzials, das als *horizontale Wiederverwendung*, also die Wiederverwendung der Muster einer Phase, charakterisiert werden kann, erschlossen. Weiteres Potenzial steckt in der Wiederverwendung von Entscheidungen des Systementwicklungsprozesses, soweit sie an die Verwendung von Mustern koppelbar sind (*vertikale Wiederverwendung*). Die Nutzung dieses Potenzials setzt jedoch die syntaktische und semantische Integration der Entwicklungsphasen voraus.

Die musterbasierte Entwicklung trägt zudem ganz wesentlich zur Verfolgbarkeit von Entwicklungsentscheidungen bei. Innerhalb einer Phase führt die Musteranwendung zu einer zusätzlichen, orthogonalen Systemstruktur, die die Art und Abfolge von Entwicklungsschritten dokumentiert (*horizontale Verfolgbarkeit*). Phasenübergreifend können darüberhinaus auch Abhängigkeiten zwischen Systemteilen verschiedener Abstraktionsstufen sichtbar gemacht werden (*vertikale Verfolgbarkeit*). Auch dies setzt die Integration der Entwicklungsphasen voraus.

3.5.2 Arbeitsprogramm

Die in Kapitel 3.5.1 beschriebenen Arbeitsschwerpunkte werden im folgenden verfeinert und erläutert. Insbesondere wird eine Abgrenzung zu den Arbeiten der Förderperioden 1 und 2 vorgenommen.

Arbeitsschwerpunkt A: Musterbasierter Protokollentwurf

In den Förderperioden 1 und 2 wurde als ein zentrales Ergebnis der SDL-Pattern-Ansatz für den musterbasierten Entwurf maßgeschneiderter Protokolle entwickelt. Bestandteile dieses Ansatzes sind ein inkrementelles Prozessmodell zur Erfassung der

Analyse- und Entwurfsmethodik sowie ein SDL-Pattern-Pool zur Dokumentation von Erfahrungswissen. Mit Ende der 2. Förderperiode kann dieser Ansatz als konsolidiert und wohlverstanden eingestuft werden. Gleichwohl gibt es eine Reihe von Punkten, an denen der Ansatz weiterentwickelt und verbessert werden kann. Da der Ansatz auf großes Interesse in Forschung und Entwicklung gestoßen ist, erscheint es lohnend, ihn weiterzuverfolgen und das erkannte Verbesserungspotenzial umzusetzen.

Arbeitspaket A1: Präzise Beschreibung von SDL-Patterns und ihrer Anwendung

Eng verknüpft mit dem Arbeitspaket A3 ist die Aufgabe der präzisen Beschreibung und der komfortablen Anwendung von Mustern. Hierzu wurden in der laufenden Förderperiode bereits Vorarbeiten geleistet, die fortgesetzt und zu einem konsolidierten Ergebnis geführt werden sollen. Insbesondere geht es darum, die Notationen zur Formulierung von SDL-Fragmenten (PA-SDL; insbesondere im Hinblick auf die Nutzung der zusätzlichen Ausdrucksmöglichkeiten von SDL-2000), Redefinitionsregeln und Einbettungsvorschriften hinreichend zu präzisieren und sämtliche SDL-Patterns in diesen Notationen zu dokumentieren. Auf dieser Grundlage sind rechnergestützte Werkzeuge zur Dokumentation und Verwaltung von SDL-Patterns in der Erfahrungsdatenbank SFB-EDB sowie zu deren Anwendung zu entwickeln.

Personalbedarf: 15 PM

Arbeitspaket A2: Definition von Analysemodellen

Teil des SDL-Pattern-Beschreibungsrahmens ist das Analysemodell, in dem ausgehend von Kommunikationsanforderungen die interagierenden Komponenten, ihre Beziehungen sowie ihre Kollaborationen definiert werden. In den Förderperioden 1 und 2 wurden zur Formulierung des Analysemodells UML-Klassendiagramme und MSC eingesetzt. Es hat sich jedoch gezeigt, dass die Sprachunterstützung hier verbesserungsfähig ist. Dazu finden in der laufenden Förderperiode bereits Vorarbeiten statt.

Um neben Verbesserungen beim Reverse-Engineering von SDL-Systemen auch den eigentlichen Entwurfsprozess zu optimieren, soll eine eigene Notation (CoSDL) entwickelt werden, die es erlaubt, initiale Entwurfsideen in Form von Kollaborationsdiagrammen zu beschreiben. CoDSL wird als graphische Notation konzipiert, die sich stark an SDL anlehnt und (ähnlich zu MSC) Interaktionsverhalten zwischen SDL-Prozessen beschreibt. Im Gegensatz zu MSC soll CoSDL aber nicht auf der Spezifikation von Beispielabläufen basieren, sondern vollständige Kollaborationen zwischen SDL-Prozessen beschreiben. Im Prozessmodell des SDL-Pattern-Ansatzes sollen Kollaborationsdiagramme während der Systemanalyse eingesetzt werden.

Personalbedarf: 6 PM

Arbeitspaket A3: Pattern Mining

Die Qualität der Mustersammlungen ist entscheidend für den Erfolg musterbasierter Entwicklungsprozesse. Muster werden in der Regel durch Analyse von Entwicklungsdokumenten abgeschlossener Projekte gewonnen, so dass von einem *Pattern Mining*

gesprochen werden kann. Dies entspricht den Schritten *analyze* und *package* des Quality Improvement Paradigms (QIP). Im Sinne dieses Ansatzes ist die Mustersammlung Bestandteil der Erfahrungsdatenbank SFB-EDB.

Ziel dieses Arbeitspakets ist die Verbreiterung und Verbesserung der bestehenden Sammlung von SDL-Patterns. Insbesondere geht es darum, weitere Protokolltypen auf Architektur- und Interaktionsmuster hin zu analysieren, die so gewonnenen Muster zu konsolidieren und sie in die Mustersammlung zu integrieren. Hierzu soll ein Improvement-Cycle zum kooperativen Pattern Mining definiert und eingesetzt werden, so dass Protokollexperten außerhalb des SFBs an der Erstellung und Verbesserung von Mustern beteiligt werden. Außerdem sollen die zusätzlichen Ausdrucksmöglichkeiten von SDL-2000 zur Musterbeschreibung genutzt werden.

Personalbedarf: 6 PM

Arbeitspaket A4: Musterbasiertes Engineering von Mikroprotokollen

Zur praktischen Erprobung des verbesserten und ergänzten SDL-Pattern-Ansatzes soll eine Menge von Mikroprotokollen (vgl. Arbeitsschwerpunkt B) unter Einsatz von Mustern entwickelt werden. Dies ist aufgrund der im Vergleich zu Mikroprotokollen feineren Granularität von SDL-Patterns sinnvoll und zweckmäßig, um auch hier zu systematischen Lösungen mit einem hohen Wiederverwendungsanteil zu gelangen. Die Mikroprotokolle sind dabei in Abstimmung mit Arbeitsschwerpunkt B so zu wählen, dass ein breiter Funktionsumfang abgedeckt wird und verschiedene Konfigurationsvarianten unterstützt werden.

Personalbedarf: 18 PM

Arbeitsschwerpunkt B: Micro-Protocol Engineering

Dieser neue Arbeitsschwerpunkt soll die Basis für die Erhöhung des Wiederverwendungsgrades bei der Entwicklung maßgeschneiderter Kommunikationsmiddleware verbreitern. Mikroprotokolle sind in sich abgeschlossene, komponierbare Protokolle mit sehr kleinem Funktionsumfang und hoher Orthogonalität. Damit handelt es sich um eine Komponententechnologie, die jedoch anders als die üblichen Vertreter bereits in der Entwurfsphase ansetzt: Mikroprotokolle werden mit formalen Techniken entwickelt und sind auf dieser Ebene bereits komponier- und simulierbar.

Arbeitspaket B1: Definition eines Beschreibungsrahmens

Zur Dokumentation von Mikroprotokollen ist ein spezielles Format, ein Micro-Protocol Description Template, zu definieren. Dieser Beschreibungsrahmen gibt zum einen die Bestandteile einer Mikroprotokolldefinition vor; zum anderen regelt er, mit welchen Sprachmitteln diese Bestandteile zu beschreiben sind. Zu den Bestandteilen gehören in jedem Fall Protokollspezifikation und -implementierung(en). Da es sich bei Mikroprotokollen im Sprachgebrauch der Wiederverwendung um *Komponenten* handelt, sind außerdem Angaben zur Selektion und Komposition erforderlich. Weitere

Angaben können sich z.B. aus der Organisation als Mikroprotokollbibliothek (als Teil der übergreifenden SFB-EDB) ergeben.

Personalbedarf: 3 PM

Arbeitspaket B2: Entwicklung einer Bibliothek von Mikroprotokolldefinitionen

Unter Verwendung des Beschreibungsrahmens aus Arbeitspaket B1 ist eine exemplarische Bibliothek von Mikroprotokolldefinitionen anzulegen. Die Bibliothek soll einen breiten Funktionsumfang abdecken und insbesondere Mikroprotokolle für den Austausch von Mess- und Stellwerten, die verteilte Regelung des Zugriffs auf ein Broadcast-Medium, Routing, Systemüberwachung sowie verschiedene Systemmanagementaufgaben beinhalten. Verschiedene Mikroprotokollkonfigurationen sind zu erstellen und zu simulieren. Die Erstellung geschieht in Koordination mit Arbeitspaket A4.

Personalbedarf: 15 PM

Arbeitspaket B3: Engineering maßgeschneiderter Kommunikationsmiddleware im SFB-Anwendungsbereich

Die Mikroprotokollbibliothek aus Arbeitspaket B2 soll zur Erstellung maßgeschneiderter Kommunikationsmiddleware für den Gebäudeautomationsbereich eingesetzt werden. Jeder Entwicklungsschritt ist durch Validationsaktivitäten abzusichern. Insbesondere sollen die konfigurierten Kommunikationssysteme auf Entwurfsebene in einem Simulationsschritt überprüft werden, bevor sie auf dem realen System implementiert werden.

Personalbedarf: 9 PM

Arbeitsschwerpunkt C: Ableitung effizienter Protokollimplementierungen

Das Wiederverwendungspotenzial der Implementierungsphase wurde in den Förderperioden 1 und 2 durch Generierungstechniken nutzbar gemacht. Zur automatischen Kodeerzeugung dient ein kommerzieller Kodegenerator in Verbindung mit einer C-Laufzeitbibliothek für QNX-Plattformen. Zur Anbindung von SDL-Spezifikationen an verschiedene Basistechnologien (CAN, ATM, Token Ring) wurde ein spezieller Kodegenerator entwickelt, der in der Lage ist, zu einer gegebenen SDL-Spezifikation die notwendigen Laufzeitroutinen automatisch zu erzeugen. Die genannten Werkzeuge wurden zur automatischen Generierung von Prototypimplementierungen in mehreren Fallstudien eingesetzt. Dabei stand vor allem die Untersuchung der Machbarkeit im Vordergrund; Speicher- und Laufzeiteffizienz wurden bisher nicht untersucht. Hier sollen zunächst die vorhandenen Resultate gesichtet und soweit möglich einbezogen werden. Insbesondere soll der kommerzielle Kodegenerator C-Micro Verwendung finden. Darüberhinaus sollen die nachfolgenden, speziellen Aspekte evaluiert werden.

Arbeitspaket C1: Untersuchung musterbasierter Implementierungstechniken

Ausgehend von einem musterbasierten Protokolldesign soll untersucht werden, inwieweit eine *musterbasierte Implementierungstechnik* zu effizienteren Implementierungen beitragen kann. Damit verbunden ist die Fragestellung, inwieweit das aus der Musteranwendung resultierende Wissen zur Optimierung des generierten Codes genutzt werden kann.

Personalbedarf: 9 PM

Arbeitspaket C2: Ableitung von Implementierungsspezifikationen

Hier soll untersucht werden, unter welchen Voraussetzungen eine signalbasierte SDL-Spezifikation in systematischer Weise auf eine *methodenbasierte Implementierung* abgebildet werden kann. Als Ansatz erscheint hier eine Umformung der Server-Struktur auf eine Activity-Thread-Struktur in Verbindung mit einem kooperativen Scheduling von Koroutinen vielversprechend. Solche Verfahren stellen sehr geringe Anforderungen an die Plattform und eignen sich u.a. auch zur Implementierung von Echtzeitsystemen. Die Festlegung einer Implementierungsstrategie erfolgt in enger Abstimmung mit Teilprojekt B5.

Im Anwendungsfeld "Gebäudeautomation" des SFB ist Skalierbarkeit ein wichtiger Entwicklungsaspekt. Dies betrifft nicht nur die Anwendungssoftware, sondern auch die Kommunikationssysteme. Demnach soll eine kleine Erweiterung oder Modifikation der Anwendung keine oder geringe Auswirkungen auf das Kommunikationssystem haben, die außerdem direkt lokalisierbar sein sollen. Zur Erreichung dieses Zielles werden Kommunikationssysteme zunächst anwendungsorientiert modularisiert. Es soll nun untersucht werden, inwieweit diese anwendungsorientierte Entwurfsspezifikation durch strukturelle Transformationen in eine Implementierungsspezifikation überführt werden kann.

Personalbedarf: 12 PM

Arbeitspaket C3: Implementierung von Mikroprotokollen im SFB-Anwendungsbereich

Die Implementierungstechniken sind auf eine Auswahl von Mikroprotokollen, wie sie in Arbeitspaket A4 mit dem SDL-Pattern-Ansatz entwickelt werden, anzuwenden. Die hierbei erzielten Ergebnisse fließen in die Mikroprotokollbibliothek ein.

Personalbedarf: 12 PM

Arbeitsschwerpunkt D: Syntaktische und semantische Integration der Entwicklungsphasen

In den Förderperioden 1 und 2 wurden musterbasierte Ansätze sowohl für die Problemanalysephase als auch die Entwurfsphase entwickelt. Sie sind insofern neuartig, als erstmalig formale Beschreibungstechniken zur Definition von Mustern verwendet

werden. Außerdem werden musterbasierte Techniken erstmals auch in der Problem-analysephase sowie für bisher nicht betrachtete Domänen untersucht. Um hier flexibel vorgehen zu können, wurde die Integration dieser Ansätze zunächst zurückgestellt. Mit der Konsolidierung der Ansätze in der zweiten Förderperiode erscheint eine Integration nun sinnvoll und geboten. Da die Ergebnisse der früheren Förderperioden sowie der Arbeitsschwerpunkte A, B und C hier einfließen, hat Arbeitsschwerpunkt D eine Klammerfunktion und leistet einen wichtigen Beitrag zur Konsolidierung der Ergebnisse.

Arbeitspaket D1: Integration der Entwicklungsprozesse

Bestandteil der musterbasierten Ansätze, die im Rahmen dieses Teilprojektes bereits entwickelt bzw. mitentwickelt wurden, sind inkrementelle Prozessmodelle zur Erfassung der Analyse- und Entwurfsmethodik. Diese Entwicklungsprozesse lassen sich prinzipiell in das SFB-Referenzprozessmodell einordnen. Allerdings weisen sie unterschiedliche Schnittstellen auf, die bei isolierter Anwendung der Ansätze zweckmäßig sind, die jedoch zur gemeinsamen Anwendung einer Harmonisierung bedürfen. Der SDL-Pattern-Ansatz geht beispielsweise davon aus, dass die Kommunikationsdienste lediglich informell beschrieben sind. Tatsächlich liegt jedoch bei vorangegangener Anforderungsanalyse eine formale, eigenschaftsorientierte Spezifikation vor. Ziel dieses Arbeitspaketes ist daher die Abstimmung sowie die syntaktische und semantische Integration der Entwicklungsprozesse, wobei besonderes Augenmerk auf die Harmonisierung der Beschreibungstechniken zu richten ist.

Personalbedarf: 12 PM

Arbeitspaket D2: Integration der Mustersammlungen

Weiterer Bestandteil der musterbasierten Ansätze sind die Mustersammlungen (Requirement-Pattern-Pool, SDL-Pattern-Pool). Aus den bereits diskutierten Gründen folgen die Musterdefinitionen eigenen Beschreibungsrahmen, die vor allem die isolierte Anwendung der Ansätze unterstützen. Die gemeinsame Anwendung der Ansätze lässt sich u.a. durch die Abstimmung dieser Beschreibungsrahmen verbessern. Beispielsweise steht zu erwarten, dass die syntaktische und semantische Integration zu einer Erschließung des bisher weitgehend ungenutzten Potenzials an vertikaler Wiederverwendung und Verfolgbarkeit führt. Methodisch lässt sich das beispielsweise durch die Aufnahme von Beschreibungsfragmenten in die Analysemodelle nachfolgender Phasen erreichen.

Personalbedarf: 12 PM

Arbeitspaket D3: Engineering maßgeschneiderter Kommunikationsmiddleware für skalierbare Gebäudeautomationssysteme

Die syntaktische und semantische Integration der musterbasierten Ansätze, verbunden mit der Nutzung der Mikroprotokollbibliotheken und der Verwendung musterbasierter Implementierungstechniken, soll im Anwendungsfeld des SFB praktisch erprobt

werden. Dabei geht es weniger um Machbarkeitsstudien, wie sie in den Förderperioden 1 und 2 im Vordergrund standen, sondern um Fallstudien realistischer Größenordnung, verbunden mit einem Baselineing, um zukünftige Verbesserungen der Prozesse und Bausteinsammlungen messbar zu machen.

Personalbedarf: 15 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausrüstung						
	B1	B2			B3	
	C1		C2		C3	
Ergänzungsausrüstung						
	A1		A2	A3	A4	
	D1 / D2				D3	
	A1 / B2			A4 / B3		
C1 / C2			C3 / D3			

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Das Teilprojekt B4 hat thematisch einen engen Bezug zur Aufgabenstellung des gesamten SFB sowie zu Teilprojekten in allen Projektbereichen. Dies findet seinen Ausdruck in den Kooperationen der zurückliegenden Förderperioden. Für die kommende Förderperiode ist vorgesehen, die Kooperation mit Teilprojekt A1 zur Einbindung des SDL-Pattern-Ansatzes in die SFB-EDB fortzusetzen und auf den Requirement-Pattern-Ansatz auszudehnen. Ferner soll das begonnene Messprogramm zur Evaluierung der musterbasierten Techniken fortgesetzt werden. Ein thematisch besonders enger Bezug besteht zu Teilprojekt B5, was sich in einer intensiven Kooperation zur Integration von Produkten und zur Vereinheitlichung der Plattform-Schnittstellen niederschlägt. Es ist vorgesehen, Mikroprotokollentwicklungen zur Erhöhung der Fehlertoleranz mit Teilprojekt B11 abzustimmen. Die bisherige sehr enge Kooperation mit Teilprojekt C1 soll im Rahmen der Weiterentwicklung der Sprachunterstützung fortgesetzt werden. Zur Modellierung der Entwicklungsprozesse wird Unterstützung von den Teilprojekten A2 und B1 erwartet. Die Realisierung prototypischer Systeme findet u.a. in dem von Teilprojekt D1 eingerichteten Testfeld statt.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	2	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
			stud. Hkr.	2	40.800	stud. Hkr.	2	40.800	stud. Hkr.	2	40.800
	zus.:	2	zus.:	4	249.600	zus.:	4	249.600	zus.:	4	249.600
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		3.000	(522)		3.000	(522)		3.000
			zusammen		3.000	zusammen		3.000	zusammen		3.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Gotzhein, Prof. Dr. (2) Müller, M., cand.-Inform. (3) Schaible, P., Dipl.-Inform.	Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	5 20 20	1.1.95 1.1.98 1.9.00	
Ergänzungsausrüstung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(4) Peper, C., Dipl.-Inform. (5) Thees, J., Dipl.-Inform. (6) N.N., stud. Hkr. (7) N.N., stud. Hkr.	Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	38,5 38,5 19 19	1.4.95 1.2.00 1.1.01 1.1.01	Ila Ila
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

- Pos. 1: Projektleitung und wissenschaftliche Betreuung
- Pos. 2: Arbeitsschwerpunkt B
- Pos. 3: Arbeitsschwerpunkt C

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

- Pos. 4: Arbeitsschwerpunkt A
- Pos. 5: Arbeitsschwerpunkt D
- Pos. 6: Entwicklungsarbeiten in den Arbeitspaketen A1, A4, B2 und B3.
- Pos. 7: Entwicklungsarbeiten in den Arbeitspaketen C1, C2, C3 und D3.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausrüstung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausrüstung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausrüstung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausrüstung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
	-	-	-
Summe	-	-	-

3.1 Allgemeine Angaben zum Teilprojekt B5

3.1.1 Thema

GeneSys – Generische Systemsoftware

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Systemsoftware

3.1.3 Leiter

Prof. Dr. Jürgen Nehmer
geb. 13.2.1942
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631-205-4020
Fax: 0631-205-3558
nehmer@informatik.uni-kl.de

Die Stelle des Leiters ist befristet ☒ ja ☐ nein, bis zum

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ ja	☒ nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ ja	☒ nein
Tierversuche	☐ ja	☒ nein
gentechnologische Untersuchungen	☐ ja	☒ nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	674,4	15	50	739,4
1998	201,6	3	0	204,6
1999	206,4	3	0	209,4
2000	104,4	3	0	107,4
Zwischensumme	1186,8	24	50	1260,8
2001	249,6	3	0	252,6
2002	249,6	3	0	252,6
2003	249,6	3	0	252,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Betriebssysteme stellen eine unverzichtbare Plattform für die Laufzeitunterstützung großer Anwendungssysteme dar. Gerade im Bereich der eingebetteten Systeme mit ihren stark divergierenden Anforderungen an Umfang und Qualität der benötigten Betriebssystemdienste besteht naturgemäß ein großer Bedarf an angepassten und hoch optimierten Lösungen. Heute verfügbare Ansätze zur Betriebssystemkonfiguration unterstützen eine derartige Maßschneiderung nur unzureichend.

Das Teilprojekt hat sich die Aufgabe gestellt, Methoden, Techniken und Werkzeuge zu entwickeln, um die Herstellung eines für eine bestimmte Anwendung maßgeschneiderten Betriebssystems in bisher nicht gekanntem Ausmaß zu unterstützen. Im Zentrum der Untersuchungen stehen hierbei nicht die Entwicklung grundlegend neuer Betriebssystemstrukturen, sondern die generische Auslegung komponentenbasierter Betriebssysteme und die werkzeuggestützte Anpassung dieser Systeme nach Anwendungsanforderungen, wobei neben funktionalen Eigenschaften insbesondere auch nichtfunktionale Anforderungen an das Betriebssystem zu berücksichtigen sind. Der wissenschaftliche Ansatz basiert auf einer Rahmenarchitektur für Betriebssysteme und einer darauf abgestimmten Komponententechnologie, die eine Wiederverwendung von Komponenten auf Kodeebene unterstützt und die flexible Maßschneiderung der Komponenten über generische Parameter unter weitgehender Einbeziehung von Generatortechniken ermöglicht. Darauf aufsetzend erlauben Verfahren zur Kodierung und Wiederverwendung von Expertenwissen die Steuerung des Entwicklungsvorgangs durch Anwendungsvorgaben.

Im Rahmen des Teilprojekts werden folgende wissenschaftliche Aufgabenstellungen bearbeitet, die für den Erfolg des Teilprojekts von zentraler Bedeutung sind:

- a) Entwicklung einer Komponententechnologie, die eine Beeinflussung der Eigenschaften einzelner Komponenten durch Belegung generischer Parameter ermöglicht und im Rahmen einer offenen Betriebssystemarchitektur die Herstellung maßgeschneiderter Systemvarianten durch Komposition aus Grundbausteinen erlaubt;
- b) Entwicklung von Generatortechniken zur anwendungsgesteuerten, automatischen Modifikation bzw. Vervollständigung von Komponentencode gemäß der Belegungen generischer Parameter;
- c) Entwicklung von Verfahren zur präzisen Erfassung von Betriebssystemanforderungen –insbesondere auch nicht-funktionaler Art, beispielsweise Zeit- oder Verteilungsaspekte betreffend– und zur Umsetzung dieser Anforderungen in Generatoranweisungen;
- d) Untersuchungen zu adaptiven Betriebssystemen, die ihren Funktionsumfang und die Qualität der angebotenen Dienste anwendungsgesteuert zur Laufzeit modifizieren können;
- e) Integration generisch ausgelegter Betriebssysteme mit den sie beschreibenden Dokumenten und den entwickelten Generatoren in einen methodisch unterstützten Softwareentwurfsprozess.

3.3 Stand der Forschung

Ansätze zur Maßschneiderung von Laufzeitplattformen im Bereich eingebetteter Systeme lassen sich bezüglich des Anpassungszeitpunktes grob in statische und dynamische Verfahren unterteilen. Unter statischer Maßschneiderung wird dabei die Anpassung eines Betriebssystems zur Übersetzungszeit verstanden, bei der neben einer Reduktion des Funktionsumfanges auf das tatsächlich benötigte Minimum feingranulare Modifikationen vorgenommen werden, um stringente funktionale und nichtfunktionale Anforderungen befriedigen zu können. Während die Entwicklung geeigneter Betriebssystemstrukturen und -implementierungen bereits weit fortgeschritten ist (siehe z.B. PURE [1], ProOSEK [8] oder VxWorks [7]), befindet man sich bei der Unterstützung der eigentlichen Maßschneiderung noch im Anfangsstadium. Mit Hilfe von dedizierten Werkzeugen (PURE, VxWorks) oder Sprachen (ProOSEK) versucht man, Betriebssystemexperten bei der Konfiguration der meist objektorientierten Betriebssystembaukästen zu unterstützen. Eine darüber hinausgehende Unterstützung von Anwendungsentwicklern ohne entsprechendes Hintergrundwissen bei der Maßschneiderung von Laufzeitplattformen stellt allerdings ein nach wie vor offenes Problem dar.

Dynamisch anpassbare Betriebssysteme ermöglichen die Erweiterung bzw. die Modifikation des Systems zur Laufzeit, woraus im Allgemeinen eine Anpassung des ausgeführten Programmkodes resultiert, die es geeignet zu unterstützen gilt. Die methodischen Grundlagen hierfür liegen im dynamischen Laden von Programmmodulen und in der objektorientierten und reflexiven Programmierung, wie sie z.B. in SPIN [4], VINO [5] und Apertos [3] eingesetzt werden. Der für eine dynamische

Anpassbarkeit erforderliche implementierungstechnische Überbau (siehe z.B. Jini [6]) erschwert allerdings den Einsatz derartiger Betriebssysteme im Bereich der eingebetteten Systeme. Alternative Ansätze wie in DREAMS [2] versuchen hier, durch angepasste Methoden und eine geeignete Einschränkung der angebotenen Unterstützung dynamische Anpassungen auch in eingebetteten Systemen zu unterstützen.

- [1] D. Beuche, A. Guerrouat, H. Papajewski, W. Schröder-Preikschat, O. Spinczyk, U. Spinczyk: *The PURE Family of Object-Oriented Operating Systems for Deeply Embedded Systems*; 2nd IEEE Intl. Symposium on Object-Oriented Real-Time Distributed Computing (ISORC' 99), Saint Malo, France, 1999
- [2] C. Ditze: *A Step towards Operating System Synthesis*; Proc. of the 5th Annual Australasian Conf. on Parallel And Real-Time Systems (PART), 1998
- [3] R. Lea, Y. Yokote, J. Itoh: *Adaptive operating system design using reflection*, Proc. of the 5th Workshop on Hot Topics in Operating Systems (HotOS-V), Washington, USA, 1995
- [4] P. Pardyak, B. Bershad: *Dynamic Binding for an Extensible System*; Proc. of the 2nd USENIX Symposium on Operating Systems Design and Implementation (OSDI), Seattle, USA, 1996
- [5] M. Seltzer, C. Small: *Self-monitoring and Self-adapting Operating Systems*; Proc. of the 6th Workshop on Hot Topics in Operating Systems, Cape Cod, USA, 1997
- [6] Sun Microsystems Inc.: *Jini Architectural Overview — Technical White Paper*, 1999; online verfügbar unter:
<http://www.sun.com/jini/whitepapers/architecture.pdf>
- [7] Wind River Systems: *VxWorks Programmer's Guide 5.4*, Edition 1, DOC-12629-ZD-01, 1999; online verfügbar unter:
http://www.wrs.com/pdf/vxworks_guide.pdf, 2000
- [8] 3SOFT: *ProOSEK: Product Description*; online verfügbar unter:
http://www.3soft.de/english/produkte/pro_osek/frame_pro_osek.html, 2000

3.4 Eigene Vorarbeiten

Die geplanten Arbeiten der dritten Förderungsperiode basieren maßgeblich auf den Ergebnissen der beiden vorangegangenen Förderungsperioden. Im Bereich der Basistechnologie für generische Systemsoftware sind mehrere Betriebssystembaukästen für eingebettete Systeme entstanden [15,17], anhand derer verschiedene Realisierungskonzepte für generische Komponenten [11] entwickelt und evaluiert wurden. Die dabei gesammelten Erfahrungen legen eine weitergehende Erforschung der Entwicklungsmethoden für generische Komponenten nahe. Grundlagen für Betrachtun-

gen hinsichtlich dynamischer Anpassungen wurden beispielsweise mit der Implementierung eines dynamisch konfigurierbaren Kommunikations-Frameworks [14] gelegt.

Zur Unterstützung der anwendungsgesteuerten Maßschneiderung von generischen Betriebssystemen wurden in der vergangenen Förderungsperiode erste Techniken, Verfahren und Werkzeuge erarbeitet. Exemplarisch sind hier die Analyseverfahren zur Extraktion von Betriebssystemanforderungen aus SDL-Anwendungsentwürfen [19,20], die Technik zur einheitlichen Modellierung von Betriebssystemressourcen [10], die Konfigurationswerkzeuge für generische Betriebssystemkomponenten [15,16,18] und der auf Design-Spaces [13] basierende Maßschneiderungsprozess [9,12] für Laufzeitplattformen zu nennen.

Mit Abschluss der zweiten Förderungsperiode stellt das Teilprojekt B5 dem Sonderforschungsbereich somit zwei maßschneiderbare Laufzeitplattformen und die entsprechenden Unterstützungswerkzeuge bereit, die im Rahmen von Fallstudien und Experimenten eingesetzt werden können.

Konferenzen und Workshops

- [9] L. Baum, M. Becker, L. Geyer, G. Molter: *Mapping Requirements to Reusable Components using Design Spaces*, Proc. of the IEEE Int'l Conference on Requirements Engineering (ICRE-2000), Schaumburg/Chicago, USA, June 2000
- [10] L. Baum, T. Kramp: *Towards a Uniform Modeling Technique for Resource-Usage Scenarios*, Proc. of the 1999 Int'l Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Las Vegas, Nevada, USA, June 1999
- [11] L. Baum: *Towards Generating Customized Run-time Platforms from Generic Components*, Proc. of the 11th Conference on Advanced Information Systems Engineering (CAISE'99), 6th DC, Heidelberg, Germany, June 1999
- [12] L. Baum, M. Becker, L. Geyer, G. Molter, P. Sturm: *Driving the Composition of Run-time Platforms by Architectural Knowledge*, Proc. of the 8th ACM SIGOPS European Workshop, Portugal, September 1998
- [13] L. Baum, L. Geyer, G. Molter, S. Rothkugel, P. Sturm: *Architecture-centric Software Development Based on Extended Design Spaces*, Proc. of the 2nd ARES Workshop (Esprit 20477), Las Palmas de Gran Canaria, February 1998

SFB-Berichte

- [14] T. Kramp, G. Coulson: *The Design of a Flexible Communications Framework for Next-Generation Middleware*, SFB 501 Report 12/99, Dept. of Computer Science, University of Kaiserslautern, 1999

Diplomarbeiten

- [15] A. Eckert: *Modellierung und werkzeuggestützte Verarbeitung von Parameterabhängigkeiten bei generischen Komponenten*; AG Systemsoftware, Universität Kaiserslautern, 1999
- [16] A. Gilbert: *Design-Space-basierte Konfiguration des Betriebssystems QNX*; AG Systemsoftware, Universität Kaiserslautern, 1999
- [17] M. Iannizzi: *Exemplarische Realisierung eines Scheduler-Subsystems für einen Realzeit-Betriebssystemkern als generische Komponente*, AG Systemsoftware, Universität Kaiserslautern, 1998
- [18] T. Leyer: *Ein Konfigurationswerkzeug für generische Komponenten am Beispiel von QNX-Modulen*, AG Systemsoftware, Universität Kaiserslautern, 1998

Projektarbeiten

- [19] E. Becher: *Modellierung von Betriebssystemanforderungen in SDL mit SDT*, AG Systemsoftware, Universität Kaiserslautern, 1998
- [20] T. See: *Erfassung von BS-Anforderungen aus SDL-Entwürfen*, AG Systemsoftware, Universität Kaiserslautern, 1998

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Aufbauend auf den in der ersten Förderungsperiode geschaffenen Grundlagen wurden in der zweiten Periode Verfahren zur statischen Maßschneiderung von Laufzeitplattformen entwickelt. Der Fokus lag hierbei auf der Unterstützung von Selektion, Konfiguration und Komposition generischer Komponenten, sowie auf der Steuerung dieser Entwicklungsschritte durch Anwendungsanforderungen. Von diesen Ergebnissen ausgehend, verfolgt das Teilprojekt in der beantragten Förderungsperiode drei Ziele. Zum einen soll für die bisher entwickelten Methoden und Werkzeuge ein durchgängiger, selbst maßschneiderbarer Prozessrahmen bereitgestellt werden. Zum anderen sollen zwei weitere, thematisch angrenzende Problemfelder aufgegriffen werden, in denen bisher keine entsprechende Unterstützung in Form von Methoden und Werkzeugen angeboten wird. Sie betreffen eine Entwicklungsunterstützung für generische Softwarekomponenten, die die Ausgangsprodukte für den bisher betrachteten Maßschneiderungsprozess darstellen, sowie Untersuchungen zur dynamischen Anpassung einer Laufzeitplattform nach deren initialer Maßschneiderung. Die sich ergebenden Arbeitsschwerpunkte fügen sich bezüglich der Entwicklungs- und Einsatzzeit einer Laufzeitplattform wie folgt in einen zeitlichen Rahmen ein (Abb. 1):

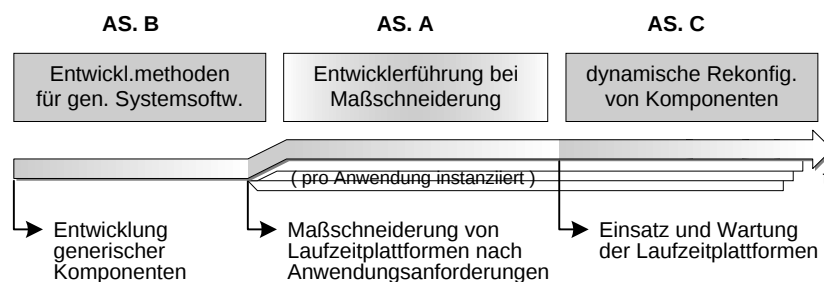


Abbildung 1: Arbeitsschwerpunkte und ihre Zuordnung zu bestimmten Phasen der Entwicklung und des Einsatzes von Laufzeitplattformen.

Im Arbeitsschwerpunkt A wird die Integration der bisher erzielten Ergebnisse bezüglich der Konstruktion maßgeschneiderter Laufzeitplattformen in einen durchgängigen Prozessrahmen angestrebt, welcher sich nahtlos in den SFB-Referenzprozess einfügt. Ziel dabei ist es, die bereitgestellten Unterstützungsleistungen von der Automatisierung einzelner Entwicklungsschritte auf die methodische Führung des Anwendungsentwicklers durch den gesamten Prozess der Maßschneidung auszudehnen. Darüber hinaus soll dieser Prozessrahmen modular strukturiert werden, um eine Maßschneidung von Entwurfsprozessen durch Einbinden spezialisierter Teilprozesse zu ermöglichen. Auf diese Weise wird auch die Integration der in den Arbeitsschwerpunkten B und C zu erzielenden Ergebnisse ermöglicht, womit letztlich ein durchgängiges Gesamtkonzept für Entwicklung, Maßschneidung und Einsatz generischer Laufzeitplattformen bereitgestellt werden kann.

Ziel des Arbeitsschwerpunktes B ist die Entwicklung von Methoden und Werkzeugen zur Unterstützung von Betriebssystementwicklern bei der Erstellung und Weiterentwicklung generischer Komponenten. Einer solchen Unterstützung kommt eine besondere Bedeutung zu, weil die Entwicklung wiederverwendbarer, generischer Komponenten im Vergleich zu einmaligen Spezialentwicklungen vergleichsweise teuer ist. Ausgehend von den bisher bei der manuellen Komponentenerstellung gewonnenen Erfahrungen sollen Techniken und Werkzeuge zur Automatisierung einzelner Teilaufgaben wie der Erstellung und Parametrisierung von Komponentenbeschreibungen entwickelt werden. Grundlage derartiger Unterstützungsleistungen können im Teilprojekt bereits eingesetzte Techniken wie die der Design Spaces sowie bestehende Generatorkonzepte bilden, die hierfür geeignet weiterzuentwickeln sind.

Im Arbeitsschwerpunkt C sollen die technischen und methodischen Grundlagen zur dynamischen Rekonfiguration aus generischen Komponenten erzeugter Laufzeitplattformen erforscht werden. Geeignete Unterstützungsleistungen, die sowohl anwendungsgesteuerte als auch durch die Plattform selbst initiierte Anpassungen von Laufzeitplattformdiensten betreffen, sind angesichts der für eingebettete Kontrollsysteme typischen langen Einsatzzeiten von besonderer Relevanz. Die bisher auf eine

statische Maßschneiderung beschränkten Generik-Konzepte müssen zu diesem Zweck um die Unterstützung ausgewählter dynamischer Anpassungen ergänzt werden. Der Ansatz generischer Komponenten ist hierbei so zu erweitern, dass generische Parameter auch nach der Instanziierung von Komponenten sichtbar bleiben und zur Anpassung der Komponenten während der Laufzeit des Systems genutzt werden können.

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Entwicklerführung bei der Maßschneiderung von Laufzeitplattformen

Im Rahmen des Arbeitsschwerpunktes A soll eine technisch und methodisch durchgängige Entwicklerunterstützung für die Maßschneiderung von Laufzeitplattformen bereitgestellt werden. Arbeitspaket A1 legt hierzu die Grundlage in Form eines konfigurierbaren Gesamtprozesses, für dessen Durchführung in Arbeitspaket A2 eine umfassende Werkzeugunterstützung bereitgestellt werden soll. Arbeitspaket A3 schließlich dient der Bewertung der entwickelten Softwaretechnologie in Bezug auf Produktivität und Produktqualität.

Arbeitspaket A1: Konfigurierbarer Gesamtprozess

Ziel dieses Arbeitspaketes ist die Erarbeitung eines Entwicklungsprozesses, der sämtliche Arbeitsschritte der statischen Maßschneiderung von Laufzeitplattformen abdeckt. Der Prozess soll mit Hilfe von MILOS formuliert werden, um die Verarbeitbarkeit durch im SFB verfügbare Werkzeuge sicherzustellen. Die Bereitstellung eines derart umfassenden Prozesses erfordert zunächst die Entwicklung von Vorgehensmodellen für bisher nicht betrachtete Bereiche wie der Verteilung einer Laufzeitplattform auf mehrere Rechnerknoten und der Auswahl einer geeigneten Hardwareplattform. Ausgehend vom Kommunikationsverhalten der Anwendung bzw. von Anwendungsanforderungen an die Hardware sind hierzu heuristische Verfahren zur Unterstützung entsprechender Entwurfsentscheidungen zu entwickeln. Darüber hinaus soll durch eine modulare Strukturierung des Prozesses die Austauschbarkeit von Teilprozessvarianten ermöglicht werden, um den Gesamtprozess für konkrete Maßschneiderungsaufgaben gezielt konfigurieren zu können. Mit dem Aufstellen von Regeln, welche die Eignung einzelner Teilprozessvarianten für bestimmte Maßschneiderungsaufgaben beschreiben, soll die Konfiguration des Gesamtprozesses unterstützt werden. Typische Anwendungsfälle und daraus resultierende Maßschneiderungsaufgaben sind hierfür zu klassifizieren, beispielsweise in Bezug auf Verteilungsanforderungen, Typen benötigter Plattformfunktionalität und Arten nichtfunktionaler Anforderungen. Ebenso sind die Teilprozessvarianten gemäß ihrer Besonderheiten und anhand der abgedeckten Prozessschritte zu charakterisieren. Gelingt eine Formulierung dieser Eigenschaften mit Design Spaces, so ließen sich geeignete Regeln in Form von Korrelationen formalisieren und einer werkzeuggestützten Verarbeitung zugänglich machen.

Personalbedarf: 12 PM

Arbeitspaket A2: Integrative Werkzeugunterstützung

Für Konfiguration und Durchführung des in Arbeitspaket A1 erarbeiteten Gesamtprozesses soll in diesem Arbeitspaket eine werkzeugtechnische Unterstützung bereitgestellt werden. Dies erfordert zum einen die Entwicklung eines Werkzeugs zur Auswahl geeigneter Teilprozessvarianten auf Basis der in Arbeitspaket A1 formulierten Regeln. Zum anderen soll ein Prozessleitwerkzeug die Durchführung des Maßschneiderungsprozesses unterstützen und die Koordination der bereits verfügbaren Einzelwerkzeuge übernehmen. Für die Bereitstellung eines solchen Werkzeugs ist auf die Ergebnisse des Teilprojekts A2 zurückzugreifen, wobei zu prüfen ist, inwieweit im Teilprojekt A2 entwickelte Werkzeuge zu erweitern sind, um die angestrebte Integration mit den Einzelwerkzeugen des Teilprojekts B5 zu ermöglichen. Das Prozessleitwerkzeug soll in diesem Zusammenhang Infrastrukturleistungen im Sinne einer offenen Werkbank zur Verfügung stellen, was beispielsweise einen einheitlichen Rahmen für Benutzeroberflächen sowie eine einheitliche Datenhaltung betrifft. In den vergangenen Förderungsperioden entstandene Werkzeuge wie der SDL-Analysator, das Design-Space-Werkzeug und der QNX-Konfigurator sind hierzu in Hinblick auf eine Angleichung der Dateiformate und Datenbankschemata, sowie eine konzeptionelle Entkopplung von Benutzeroberfläche und Werkzeugfunktionalität anzupassen. Auf diese Weise kann das Prozessleitwerkzeug neben der Führung des Entwicklers durch den gesamten Maßschneiderungsprozess auch die zeitlich passende Ansteuerung der Einzelwerkzeuge und die Verwaltung teilprozessübergreifender Statusinformationen übernehmen.

Personalbedarf: 15 PM

Arbeitspaket A3: Kosten-/Nutzenbewertung

Mit Hilfe von Fallstudien soll in diesem Arbeitspaket eine Bewertung von Kosten und Nutzen des verfolgten Entwicklungsansatzes sowie der bereitgestellten Werkzeugunterstützung vorgenommen werden. Die Untersuchungen sollen sich dabei insbesondere auf die zu erzielende Produktivitätssteigerung bei der Entwicklung maßgeschneiderter Laufzeitplattformen und auf die erreichbare Produktqualität konzentrieren, wobei als Standardmetriken Entwicklungszeit und Fehleranzahl zum Einsatz kommen. Während sich die Untersuchungen in der vergangenen Förderungsperiode ausschließlich auf die Nutzenseite bezogen, soll hier auch die Kostenseite betrachtet werden. In einzelnen Fallstudien ist daher mit Unterstützung des Teilprojekts A1 zum einen der Entwicklungsaufwand für ausgewählte generische Komponenten zu quantifizieren und mit dem Aufwand für konventionelle Komponentenentwicklungen zu vergleichen. Zum anderen sind in Fallstudien beispielsweise auf Basis der Plattformarchitektur RTK Entwicklungen maßgeschneiderter Plattformen sowohl manuell als auch mit Hilfe der zur Verfügung stehenden Werkzeugunterstützung vorzunehmen, wobei Entwicklungszeit und Fehleranzahl zu erfassen sind. Auf Grundlage dieser Daten ist erstmals eine Gesamtbewertung des Entwicklungsansatzes möglich, beispielsweise im Sinne von Aussagen, ab welcher Wiederverwendungshäufigkeit sich der Einsatz generischer Laufzeitplattformen lohnt.

Personalbedarf: 12 PM

Arbeitsschwerpunkt B: Entwicklungsmethoden für generische Systemsoftware

In diesem Arbeitsschwerpunkt sollen Verfahren zur Erstellung generischer Komponenten und maßschneiderbarer Plattformen erforscht und durch Werkzeuge unterstützt werden. Die Arbeitspakete B1 und B2 beschäftigen sich hierbei mit komponentenlokalen Fragestellungen, die Arbeitspakete B3 und B4 mit komponentenübergreifenden und architekturbezogenen Fragestellungen.

Arbeitspaket B1: Komponentenentwicklung

Ziele dieses Arbeitspaketes sind die Erarbeitung eines Vorgehensmodells für die Entwicklung generischer Komponenten und die Bereitstellung einer auf das Konzept generischer Komponenten abgestimmten Entwicklungsumgebung. Ausgehend von den in vergangenen Förderungsperioden gesammelten Erfahrungen mit ad-hoc-Entwicklungen generischer Komponenten sollen Heuristiken erarbeitet werden, welche beispielsweise Aussagen darüber zulassen, wieviel Flexibilität an welchen Implementierungsstellen sinnvoll ist und mit welchen der entwickelten Implementierungstechniken die jeweilige Flexibilität optimal realisiert werden kann. Darüber hinaus gilt es zu untersuchen, wie die bisher in unterschiedlichen Formaten wie key-value-pairs, sed-Skripten, und Quelltextkommentaren vorliegenden Parametrisierungsinformationen vereinheitlicht werden können, um sie einfacher für Entwicklungswerkzeuge verfügbar zu machen. Als Basis für eine einheitliche Beschreibung der Parametrisierung sowohl des Quelltextes als auch der VA-APIs bietet sich die Sprache XML an, die eine leichte Transformierbarkeit der Beschreibungen in diverse andere Formate verspricht. Die zur Unterstützung dieses Ansatzes erforderliche Entwicklungsumgebung schließlich soll eine einfache Eingabe von Komponentenbeschreibungen und insbesondere die Bearbeitung von Komponentenquelltexten in verschiedenen Sichten ermöglichen. Eine Aufbereitung des Quelltextes z.B. für eine Generik-Sicht auf alle Parametrisierungsinformationen oder eine Instanzen-Sicht auf eine einzelne ausgewählte Variante bietet sich auf Basis einer Transformation in HTML an.

Personalbedarf: 15 PM

Arbeitspaket B2: Generierung von Komponentenbeschreibungen

Die Entwicklung von Komponentenbeschreibungen und insbesondere die Bereitstellung von Parametrisierungsinformationen kann bis zu einem gewissen Grad durch Werkzeuge unterstützt werden. Im Zentrum dieses Arbeitspaketes stehen daher Untersuchungen, inwieweit derzeit für die Manipulation von Komponentenimplementierungen eingesetzte, auf Java-Macros und XML basierende Generatortechniken zu erweitern sind, um einzelne Beschreibungsfacetten generischer Komponenten automatisch zu erzeugen. Mittels geeignet angepasster Generatoren lassen sich beispielsweise Schnittstellenbeschreibungen automatisch aus dem Komponentenquelltext ableiten, wobei im Falle varianter Schnittstellen gleichzeitig auch entsprechende Parametrisierungsinformationen generiert werden können. Im Falle mehrerer konventionell implementierter Komponentenvarianten lassen sich darüber hinaus unter bestimmten Umständen generische Implementierungen mitsamt der notwendigen generischen Parameter und der Generierungsinformationen erzeugen. Liegen die ein-

zelen Varianten zu diesem Zweck in strukturell ähnlicher Form vor, beispielsweise weil sie durch geringfügige Modifikationen aus dem selben Quelltext hervorgegangen sind, so lassen sich durch zeilenweisen Quelltextvergleich Informationen über invariante und variante Komponententeile gewinnen. Diese Informationen können automatisch zur Ableitung generischer Parameter und zur Erzeugung einer generischen Implementierung genutzt werden. Neben einer nachträglichen generischen Auslegung bereits vorhandener konventioneller Komponenten wird auf diese Weise auch die Weiterentwicklung generischer Komponenten unterstützt.

Personalbedarf: 12 PM

Arbeitspaket B3: Ableitung von Architekturinformationen

Bei der Entwicklung neuer Komponenten bzw. Komponentenvarianten müssen deren Wechselwirkungen mit anderen Elementen der Zielarchitektur erkannt und beschrieben werden. Dieses Arbeitspaket dient daher der Entwicklung von Methoden zur teilautomatischen Erfassung bestimmter architektureller Informationen wie Komponentenabhängigkeiten und Kompositionsregeln während der Komponentenentwicklung. Abhängigkeiten in Form von Aufrufen, Datenzugriffen und Referenzierungen lassen sich bereits während der Implementierung der Komponenten erkennen und werkzeuggestützt beispielsweise in Form annotierter Korrelationen zwischen Komponenten-Design-Spaces dokumentieren. Zur automatischen Erfassung derartiger Informationen sind geeignete Quelltext-Parser bereitzustellen und in die angestrebte Komponentenentwicklungsumgebung zu integrieren. Darüber hinaus gilt es, mit Unterstützung des Teilprojekts C1 zu untersuchen, inwieweit sich diese Abhängigkeiten auf Basis eines geeignet erweiterten Korrelationskonzepts in formale Kompositionsregeln für Komponenten transformieren lassen. Einen ersten Ansatzpunkt stellen hierbei erweiterte Abstract State Machines (ASM) dar. Für eine nahtlose Integration in die bereits entwickelte Methodik zur Auswahl und Konfiguration von Komponenten muss sich eine entsprechende Formalisierung des Kompositionsbegriffs dabei mit dem Design-Space-Konzept kombinieren lassen.

Personalbedarf: 12 PM

Arbeitspaket B4: Auswahl und Evolution von Architekturen

Bevor für eine konkrete Anwendung passende generische Systemsoftwarekomponenten ausgewählt und maßgeschneidert werden können, muss zunächst eine geeignete Plattformarchitektur gewählt werden. In diesem Arbeitspaket sollen daher zum einen Methoden entwickelt werden, die die Unterstützung einer solchen Architekturauswahl durch Werkzeuge ermöglichen. Hierbei gilt es, den Einfluss architektureller Konzepte auf die Systemeigenschaften zu erfassen; Eine Beschreibung dieser Zusammenhänge soll mit Hilfe von Erweiterungen an Architekturbeschreibungssprachen wie Darwin oder Rapide versucht werden. Auf Basis dieser Erkenntnisse lassen sich dann Aussagen über die Erfüllbarkeit bestimmter Anwendungsanforderungen mit unterschiedlichen Plattformarchitekturen machen. Dieses Wissen soll in Form von Heuristiken formuliert und zur werkzeugunterstützten Steuerung der Architekturauswahl durch Anwendungsanforderungen genutzt werden. Zum anderen sollen in diesem Arbeits-

paket auch Verfahren bereitgestellt werden, die die evolutionäre Weiterentwicklung einer Architektur im Falle fehlender Funktionalitäten oder unerfüllbarer nichtfunktionaler Anforderungen unterstützen. Hierbei ist insbesondere an die Auswertung architektureller Informationen zum Auffinden durch notwendige Änderungen betroffener Komponenten gedacht. Zu diesem Zweck sind Werkzeuge zur Analyse der in Arbeitspaket B3 bereitgestellten Abhängigkeitsbeschreibungen zu entwickeln.

Personalbedarf: 12 PM

Arbeitsschwerpunkt C: Dynamische Rekonfiguration von Komponenten

Die Weiterentwicklung bestehender Generik-Konzepte zur Unterstützung dynamischer Anpassungen von Laufzeitplattformen ist Gegenstand dieses Arbeitsschwerpunktes. Die Grundlagen hierzu werden mit den Arbeitspaketen C1 und C2 gelegt. Darauf aufbauend sollen im Arbeitspaket C3 exemplarisch erste dynamisch maßschneiderbare Komponenten und entsprechende Unterstützungswerkzeuge entwickelt werden.

Arbeitspaket C1: Untersuchung von Anforderungsszenarien

Mit diesem Arbeitspaket sollen die Anforderungen an eine dynamische Anpassbarkeit von Laufzeitplattformen für eingebettete Gebäudeautomationssysteme bestimmt werden. Beispiele betreffen die Veränderung von Ressourcenverteilungen, den Wechsel von Ressourcenvergabe-strategien und die Unterstützung neuer Sensor- und Aktuatorinstanzen. In Zusammenarbeit mit den Teilprojekten D1 und D2 sind unterschiedliche Anpassungsszenarien zu untersuchen und bezüglich ihrer Anforderungen an eine Systemunterstützung zu klassifizieren. Charakteristika in diesem Zusammenhang sind beispielsweise die Zeitpunkte, zu denen die Anpassungen vorgenommen werden können bzw. müssen, der für eine Anpassung tolerierbare Zeit- und Speicherplatzbedarf, und die Frage, welche Systemkomponenten von den Anpassungen betroffen sind. Für die ermittelten Klassen dynamischer Anpassungen sind daraufhin die implementierungstechnischen Voraussetzungen zu untersuchen, z. B. Vorkehrungen für Indirektionen beim Aufruf von Komponenten oder die Bereitstellung von Infrastrukturdiensten wie Namensdiensten und dynamischen Ladern. Hauptaugenmerk soll hierbei auf Anpassungen liegen, die lediglich die Modifikation oder den Austausch einzelner Komponenten erfordern. Bekannte Ansätze zur Unterstützung dynamischer Komponentenumgebungen wie beispielsweise Jini sind zu beachten, bedürfen aber grundsätzlicher Anpassungen, um mit den strikten Speicherplatz- und Zeitanforderungen eingebetteter Systeme verträglich zu sein.

Personalbedarf: 12 PM

Arbeitspaket C2: Erweiterung der Generik-Konzepte

Ausgehend von den in Arbeitspaket C1 erfassten Anforderungen befasst sich dieses Arbeitspaket mit Fragestellungen, wie die in den vergangenen Förderungsperioden erarbeiteten Generik-Konzepte weiterzuentwickeln sind, um auch dynamische Anpassungen abzudecken. Die notwendigen Erweiterungen betreffen hierbei die Menge generischer Parameter ebenso wie deren Beschreibungsformen, erforderliche Imple-

mentierungstechniken und Generatoren. Für die in Arbeitspaket C1 identifizierten Typen von Anpassungen müssen neue Typen generischer Parameter eingeführt werden, für die auch geeignete Beschreibungskonzepte zu entwickeln sind. Als Ausgangsbasis bieten sich XML-basierte, separat zu den Komponenten vorgehaltene Beschreibungen an, da im Gegensatz zum statischen Maßschneiderungsfall Parametrisierungsinformationen nicht mehr im Quelltext verankert werden können. Für die neuartigen Parametertypen sind desweiteren geeignete Implementierungstechniken zu finden. Beispielsweise müssen für Indirektionen bei komponentenübergreifenden Aufrufen möglichst effiziente Implementierungen gefunden werden, um den Laufzeit-overhead für dynamisch austauschbare Komponenten gering zu halten. Schließlich sollen Methoden erforscht werden, welche analog zum statischen Maßschneiderungsfall die Belegung generischer Parameter in Abhängigkeit von konkreten Anforderungen unterstützen.

Personalbedarf: 12 PM

Arbeitspaket C3: Prototypische Realisierung

Die in den Arbeitspaketen C1 und C2 erarbeiteten Konzepte sollen in diesem Arbeitspaket prototypisch umgesetzt werden. Ziel ist die Entwicklung sowohl einzelner dynamisch anpassbarer Komponenten, als auch der nötigen Laufzeitplattforminfrastruktur und geeigneter Werkzeuge zur Unterstützung dynamischer Anpassungen. Die Rahmenarchitektur für die Komponenten stellt die RTK-Plattform zur Verfügung, die zu diesem Zweck um die erforderlichen Infrastrukturdienste wie Laufzeitkonfigurationsmanagement und dynamischer Lader zu erweitern ist. Für diese Architektur sind daraufhin exemplarisch dynamisch anpassbare Komponenten zu entwickeln. Beispiele betreffen Threadmanagement-Komponenten, die zur Laufzeit die Veränderung bzw. den Austausch der Schedulingstrategie erlauben, Speicherverwaltungskomponenten, die eine Veränderung der Zuteilungsstrategie ermöglichen, oder Kommunikationskomponenten, die zur Unterstützung dedizierter Kommunikationsmedien ausgetauscht werden können. Zur Initiierung und Durchführung derartiger Anpassungen sind geeignete Werkzeuge zur Verfügung zu stellen. Anhand von Fallstudien, in denen auf Basis der entwickelten Komponenten Laufzeitplattformen für Kontrollsystemanwendungen zur Verfügung gestellt werden, sollen die Praktikabilität des verfolgten Ansatzes gezeigt und erste Erfahrungen bezüglich der erzielbaren Laufzeiteffizienz gewonnen werden.

Personalbedarf: 12 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.	
Grundausrüstung V. Tamara							
	C1		C2		C3		
Ergänzungsausrüstung L. Baum M. Becker Stud. Hilfskräfte: N.N.* N.N.							
	A1		A2				
					A3		
	B1						
		B2		B3		B4	
	B1 / A2						
	C1		C2		C3		

* Mitarbeit bei der Werkzeugentwicklung in den Arbeitspaketen B1 und A2 je nach Bedarf.

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Das Teilprojekt B5 bildet zusammen mit A3, B4 und B10 diejenige Gruppe von Teilprojekten, die an generischen Produktmodellen arbeiten. Deshalb gibt es unter diesen Teilprojekten einen ständigen Erfahrungsaustausch, insbesondere im Bereich der Produkt-Modellierungstechniken. Da die Teilprojekte B4 und B5 kommunikations- und betriebssystemspezifische Teile generischer Laufzeitplattformen für Anwendungssysteme entwickeln, ist hier eine besonders intensive Kooperation zur Integration von Produkten und zur Vereinheitlichung der Plattform-Schnittstellen erforderlich. Gemeinsam mit dem Teilprojekt C1 wird an der Formalisierung der eingesetzten Beschreibungstechniken gearbeitet, was beispielsweise Design Spaces und VA-APIs betrifft. Für die Modellierung von Entwicklungsprozessen und den Rückgriff auf bereits entwickelte Prozessunterstützungswerkzeuge wird Hilfestellung von den Teilprojekten A2 und B1 erwartet.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	1	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
			StdHkr.	2	40.800	StdHkr.	2	40.800	StdHkr.	2	40.800
	zus.:	1	zus.:	4	249.600	zus.:	4	249.600	zus.:	4	249.600
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		3.000	(522)		3.000	(522)		3.000
			zusammen		3.000	zusammen		3.000	zusammen		3.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Nehmer, J.; Prof. Dr.	Prakt. Inf.	FB Inf., Uni KL	5	1.1.95	
	(2) Schröder-Preikschat, W., Prof. Dr.	Prakt. Inf.	FB Inf., Uni Magdeburg	B	1.1.98	
	(3) Tamara, V., M. Sc.	Prakt. Inf.	FB Inf., Uni KL	20	1.10.99	
Ergänzungsausstattung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Baum, L., Dipl.-Inform.	Prakt. Inf.	FB Inf., Uni KL	40	1.1.96	IIa
	(2) Becker, M., Dipl.-Inform.	Prakt. Inf.	FB Inf., Uni KL	40	1.1.98	IIa
	(3) N.N., StdHkr.	Prakt. Inf.	FB Inf., Uni KL	20	1.1.01	
	(4) N.N., StdHkr.	Prakt. Inf.	FB Inf., Uni KL	20	1.1.01	
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

- Pos 1: Projektleitung und wissenschaftliche Betreuung
- Pos 2: Wissenschaftliche Beratung
- Pos 3: Arbeitsschwerpunkt C

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

- Pos 4: Arbeitsschwerpunkt A
- Pos 5: Arbeitsschwerpunkt B
- Pos 6: Implementierungsarbeiten in Arbeitsschwerpunkten A und B
- Pos 7: Implementierungsarbeiten im Arbeitsschwerpunkt C

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausrüstung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausrüstung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausrüstung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausrüstung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
	-	-	-
Summe	-	-	-

3.1 Allgemeine Angaben zum Teilprojekt B10

3.1.1 Thema

Anwendungsentwicklung mit vorkonfektionierten Systemen

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Systemsoftware

3.1.3 Leiter

Prof. Dr. Jürgen Nehmer
geb. 13.2.1942
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-4020
Fax: 0631/205-3558
nehmer@informatik.uni-kl.de

Die Stelle des Leiters ist befristet ☒ (x) nein ☐ () ja, bis zum

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ () ja	☒ (x) nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ () ja	☒ (x) nein
Tierversuche	☐ () ja	☒ (x) nein
gentechnologische Untersuchungen	☐ () ja	☒ (x) nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
1998	100,8	1,5	0	102,3
1999	103,2	1,5	0	104,7
2000	104,4	1,5	0	105,9
Zwischensumme	308,4	4,5	0	312,9
2001	249,6	3,0	0	252,6
2002	249,6	3,0	0	252,6
2003	249,6	3,0	0	252,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Das Teilprojekt B10 hat sich die Aufgabe gestellt, die Entwicklung von Anwendungssoftware durch eine weitreichende Wiederverwendung domänenspezifischer Softwareartefakte zu unterstützen. Die in der ersten Förderungsperiode (1998-2000) gesammelten Erfahrungen zeigen, dass in Anwendungsfeldern mit potenziell großem Lösungsraum entsprechende Konzepte schon in frühe Phasen der Softwareentwicklung integriert werden müssen, um systematische Wiederverwendung zu unterstützen. Im Teilprojekt B10 wird dazu der Ansatz verfolgt, geeignete Softwarearchitekturen als Träger für wiederverwendbares Domänenwissen sowie als Rahmen für alle Wiederverwendungsaktivitäten einzusetzen. Die Kombination einer solchen Architektur mit dazu kompatiblen wiederverwendbaren Produkten und einem speziell angepassten Entwicklungsprozess stellt ein vorkonfektioniertes System dar, das im Laufe der Anwendungsentwicklung ergänzt und auf die spezifische Aufgabenstellung zugeschnitten wird.

Im Gegensatz zu anderen produktorientierten Teilprojekten im SFB 501, die sich auf die Entwicklung von Laufzeitplattformen sowie auf die dabei auftretenden Interdependenzen mit der Anwendungsentwicklung konzentrieren (B4, B5), zielen die in B10 erarbeiteten Ansätze auf die Unterstützung der gesamten Anwendungsentwicklung ab. Dazu wird ein generischer, domänenunabhängiger Technologierahmen entwickelt, der die grundlegende Vorgehensweise sowie Techniken zur Anwendungsentwicklung mit vorkonfektionierten Systemen beschreibt.

Ein einzelnes vorkonfektioniertes System stellt eine Instanz dieses Rahmens für ein spezifisches Anwendungsfeld dar. Als exemplarisches Anwendungsfeld wird hierfür in B10 die Domäne Gebäudeautomation betrachtet.

Im Rahmen des Teilprojektes werden folgende wissenschaftliche Aufgaben bearbeitet:

- Bereitstellung eines Technologierahmens für wiederverwendungsorientierte, architekturbasierte Softwareentwicklung
- Erarbeitung eines Prozessmodells für architekturzentrierte Softwareentwicklung, das auf systematische Wiederverwendung unterschiedlicher Arten von Produkten ausgelegt ist (z.B. Modellfragmente und Softwarekomponenten)
- Entwicklung von Mess- und Analysemethoden zur Überwachung der Anwendungsentwicklung unter Ausnutzung wiederverwendbaren Wissens
- Unterstützung der Entwicklung vorkonfektionierter Systeme für noch nicht erschlossene Domänen, sowie der Wartung und Weiterentwicklung existierender vorkonfektionierter Systeme

3.3 Stand der Forschung

In diesem Abschnitt werden Arbeiten vorgestellt, die einen maßgeblichen Einfluss auf das Arbeitsprogramm des Teilprojektes in der nächsten Förderungsperiode haben.

Das Konzept vorkonfektionierter Systeme bietet die Möglichkeit zur Realisierung einer Software-Produktlinie, wie sie beispielsweise in [2] beschrieben wird. Ein solcher Ansatz für die Entwicklung von Produktlinien wird mit PuLSE [1] ebenfalls am Fraunhofer Institut für experimentelles Software Engineering untersucht, mit dem diesbezüglich ein reger Gedankenaustausch erfolgt.

Der in B10 verwendete sichtenbasierte Ansatz zur Architekturbeschreibung ähnelt dem in [4] beschriebenen Architekturkonzept sowie dem von Kruchten vorgestellten „4+1 view model of architecture“ [5]. Zur Repräsentation der einzelnen Sichten können unterschiedliche Beschreibungstechniken verwendet werden, darunter verschiedene UML-Diagrammtypen [6] sowie Architekturbeschreibungssprachen wie Darwin [7].

Architekturanalyse anhand der SAAM-Methode [3] nimmt eine zentrale Rolle bei der Unterstützung von Anwendungsentwicklung mit vorkonfektionierten Systemen ein. Diese szenariobasierte Analysemethode wird hier beispielsweise zur frühzeitigen Evaluierung architektureller Entscheidungen eingesetzt.

- [1] J. Bayer, O. Flege, P. Knauber, R. Laqua, D. Muthig, K. Schmid, T. Widen, J.-M. DeBaud: *PuLSE: A Methodology to Develop Software Product Lines*; Fifth Symposium on Software Reusability 1999, ACM Press 1999
- [2] D. Weiss: *Software Product-Line Engineering: A Family-Based Software Development Process*; Addison Wesley, 1999
- [3] R. Kazman, G. Abowd, L. Bass, P. Clements: *Scenario-Based Analysis of Software Architecture*; IEEE Software, November 1996

- [4] L. Bass, P. Clements, R. Kazman: *Software Architecture in Practice*; Addison-Wesley, 1998
- [5] P. Kruchten: *The 4 + 1 View Model of Architecture*; IEEE Software, 12(6), pp. 42-50, November 1995
- [6] J. Rumbaugh, I. Jacobsen, G. Booch: *The Unified Modeling Language Reference Manual*; Addison-Wesley Object Technology Series, 1999
- [7] Jeff Magee, Naranker Dulay, Susan Eisenbach, Jeff Kramer: *Specifying Distributed Software Architectures*; Proc. of the 5th European Software Engineering Conference (ESEC '95), Sitges, September 1995, Springer LNCS 989, pp. 137-153, 1995

3.4 Eigene Vorarbeiten

Die geplanten Arbeiten bauen direkt auf den Erkenntnissen der vorangegangenen Förderungsperiode bezüglich der Prinzipien zur Gestaltung und Strukturierung vorkonfektionierter Systeme auf. Diese Aspekte wurden in mehreren Fallstudien in den Anwendungsfeldern Gebäudeautomation [15][16] und Multimedia-Server [14] untersucht.

Vorkonfektionierte Systeme zeichnen sich im Wesentlichen durch zwei Merkmale aus, zum einen durch die zentrale Rolle von Architektur insbesondere zur Unterstützung von Wiederverwendung [9][10], zum anderen durch die mit jedem vorkonfektionierten System verknüpfte spezifische Vorgehensweise, die aus einer abstrakten Sicht in [8] beschrieben wurde.

Die Arbeiten der beantragten Förderungsperiode gründen auf verschiedene in der letzten Periode entwickelte Techniken. Eine zentrale Bedeutung kommt beispielsweise dem Design Space-Konzept zu [11]. Für Validierungszwecke und zur Umsetzung architekturellen Wissens wird die erweiterte SAAM-Methode [12] verwendet. Als Grundlage für die Durchführung von Domänenanalysen und die Beschreibung von Domänenmodellen dienen die Ergebnisse des Querschnittsprojekts Q3 [13].

Konferenzen und Workshops

- [8] L. Baum, M. Becker, L. Geyer, G. Molter: *A Process View on Architecture-Based Software Development*; 1st Working IFIP Conference on Software Architecture (WICSA-1), San Antonio, Texas, USA, 22.-24. Feb. 1999
- [9] L. Baum, M. Becker, L. Geyer, G. Molter: *The Role of Architecture for Complex Systems Development*; 11th. Int'l Conf. on Software Engineering and its Applications 1998, Paris, France, 8.-10. Dec. 1998

- [10] L. Baum, M. Becker, L. Geyer, G. Molter: *Using Software architecture as a Catalyst for Reuse*; European Reuse Workshop 1998, Madrid, Spain, 4.-6. Nov. 1998
- [11] L. Baum, L. Geyer, G. Molter, S. Rothkugel, P. Sturm: *Architecture-Centric Software Development Based on Extended Design Spaces*; ARES Second Int'l Workshop on Development and Evolution of Software Architectures for Product Families, Las Palmas de Gran Canaria, Spain, Feb. 1998
- [12] G. Molter: *Integrating SAAM in Domain-Centric and Reuse-based Development Processes*; 2nd Nordic Workshop on Software Architecture (NOSA), Ronneby, Sweden, 12. - 13. August 1999

Technische Berichte

- [13] L. Geyer, M. Kronenburg, R. Merz, B. Schürmann: *Domain Modeling in the SFB 501*, SFB 501 Report 07/00, Dept. of Computer Science, University of Kaiserslautern, 2000

Diplomarbeiten

- [14] F. Hütter: *Ein vorkonfektioniertes System für echtzeitfähige Multimedia-Server*; 1998
- [15] S. Bötzel: *Eine Design Space-Klassifikation und Realisierung eines marktba-
sierten vorkonfektionierten Systems zur Gebäudeautomation*; 1998
- [16] S. Albus: *Eine Design Space-Klassifikation und Realisierung eines eventge-
steuerten vorkonfektionierten Systems zur Gebäudeautomation*; 1998

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Das Kernziel der Arbeiten in der neuen Förderungsperiode ist die Bereitstellung eines durchgängigen Technologierahmens für architekturbasierte, wiederverwendungsorientierte Softwareentwicklung. Der Ansatz ist insbesondere darauf ausgerichtet, Produkt-Wiederverwendung systematisch durch Nutzung ebenfalls wiederverwendbarer Architekturen zu unterstützen. Die Ergebnisse der vergangenen Förderungsperiode (1998-2000) – insbesondere das erarbeitete Strukturierungskonzept für vorkonfektionierte Systeme, das sichtenbasierte Beschreibungskonzept für Architekturen, sowie die bereits realisierten Techniken und Werkzeuge – dienen dabei als Grundlage für die geplanten Forschungsarbeiten.

Die Aktivitäten in der beantragten Förderungsperiode gliedern sich in drei Schwerpunkte. Zunächst soll ein Gesamtkonzept zur Anwendungsentwicklung mit vorkonfektionierten Systemen erarbeitet werden, das die bisher erzielten Ergebnisse integriert und einen Rahmen für ihre Weiterentwicklung bildet. Während die bisher-

gen Untersuchungen beispielhaft auf das Anwendungsfeld Gebäudeautomation abzielen, soll in einem weiteren Arbeitsschwerpunkt die Technologie für die systematische Erschließung weiterer Anwendungsfelder bereitgestellt werden. Desweiteren sollen architekturbasierte Mess- und Analysemethoden für Planung und Kontrolle der Entwicklung neuer vorkonfektionierter Systeme und ihrer Verwendung bei der Realisierung konkreter Anwendungen erforscht werden. Im Überblick ergibt sich damit die in Abbildung 1 dargestellte Untergliederung des Arbeitsprogramms:

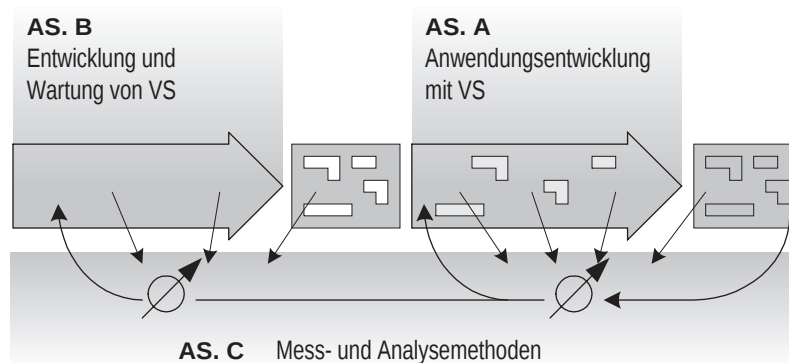


Abbildung 1: Übersichtsdarstellung der Arbeitsschwerpunkte

In Arbeitsschwerpunkt A soll eine durchgängige Entwicklungsumgebung für die architekturbasierte, wiederverwendungsorientierte Anwendungsentwicklung auf Basis vorkonfektionierter Systeme bereitgestellt werden. Die Grundlage dafür stellen die in der vergangenen Förderungsperiode untersuchten und für Teilbereiche des Anwendungsfeldes Gebäudeautomation instanziierten Ansätze und Modelle dar. Sie sollen zu einem umfassenden Produkt- und Prozessmodell weiterentwickelt werden, das als Rahmen für die systematische Anwendung vorkonfektionierter Systeme in einer Anwendungsdomäne verwendet werden kann. Für die verschiedenen Aktivitäten im Entwicklungsprozess ist eine weitreichende Werkzeugunterstützung vorgesehen.

In Arbeitsschwerpunkt B sollen anwendungsfeldübergreifende Verfahren zur Entwicklung und Wartung vorkonfektionierter Systeme erarbeitet werden. Die Entwicklung solcher Systeme erfordert die Umsetzung der Ergebnisse einer Domänenanalyse in eine Architektur, wobei besonderes Augenmerk auf die Vorgehensweise zur Modellierung der Flexibilität in der Domäne gelegt werden muss. Desweiteren ist eine Domäne nach der Analyse typischerweise Veränderungen unterworfen; infolgedessen muss die Architektur des vorkonfektionierten Systems angepasst werden. Diese Änderungsanforderungen werden in der Regel bei Instanziierungen des vorkonfektionierten Systems entdeckt; als Reaktion muss ein Wartungsprozess gestartet werden, in dem die entsprechenden Veränderungen vorgenommen werden.

Arbeitsschwerpunkt C umfasst die Entwicklung von Mess- und Analysetechniken für den Einsatz sowohl bei der Anwendungsentwicklung mit vorkonfektionierten Systeme-

men (Arbeitsschwerpunkt A) als auch bei der Bereitstellung von vorkonfektionierten Systemen für ein bestimmtes Anwendungsfeld (Arbeitsschwerpunkt B). Je nach Kontext werden dabei jedoch unterschiedliche Anforderungen an die Analyseergebnisse gestellt:

- So sollen es die zu realisierenden Techniken erlauben, bei der Anwendungsentwicklung frühzeitig Prognosen zum einen des erforderlichen Entwicklungsaufwandes, zum anderen der Eigenschaften der fertigen Anwendung zu erstellen.
- Im Rahmen der Erstellung von vorkonfektionierten Systemen werden die Techniken in einem Verifikationsschritt dazu verwendet, die korrekte Umsetzung der Ergebnisse der Domänenanalyse in das System sicherzustellen. Besonderes Augenmerk liegt dabei auf der Analyse der Flexibilität der entwickelten Architektur.
- Bei der Wartung schließlich bilden die Analyseergebnisse die Grundlage für gegebenenfalls notwendige Anpassungen der Architektur an geänderte bzw. erweiterte Anforderungen in der Domäne.

Die in diesem Arbeitsschwerpunkt entwickelten Mess- und Analysemethoden sollen in einem weiteren Schritt in die in den Arbeitsschwerpunkten A und B bereitgestellten Modelle integriert werden.

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Anwendungsentwicklung mit vorkonfektionierten Systemen

Arbeitspaket A1: Produktmodell für Anwendungsentwicklung mit vorkonfektionierten Systemen

Eine der wesentlichen Grundlagen für die systematische Unterstützung der Anwendungsentwicklung mit vorkonfektionierten Systemen ist die Bereitstellung eines Rahmens für die Beschreibung aller Produkte, die in solchen Systemen enthalten sein können. In diesem Arbeitspaket soll daher ein generisches Produktmodell für vorkonfektionierte Systeme erarbeitet werden, das die Grundlage für die Modellierung von Produkten und Entwicklungsprozessen für spezifische vorkonfektionierte Systeme bietet.

Den Ausgangspunkt für diese Modellierungsarbeiten bilden das in der ersten Förderungsperiode erarbeitete Zuordnungsschema von wiederverwendbaren Artefakten zu Entwicklungsaktivitäten, sowie die realisierten vorkonfektionierten Systeme [14][15][16] für verschiedene Anwendungsfelder. Die in ihnen enthaltenen verschiedenen Typen von Produkten sind zunächst zu klassifizieren und in einen systematischen Zusammenhang zu bringen. Insbesondere ist auf eine offene Gestaltung des Produktmodells Wert zu legen, um seine Anwendbarkeit für unterschiedliche Domänen sicherzustellen. Dazu sind entsprechende abstrakte Produktklassen zu definieren, die es erlauben, für einzelne vorkonfektionierte Systeme spezifische Produkte zu inte-

grieren. Eine solche Produktklasse bilden beispielsweise die verschiedenen Strukturen und Sichten zur Architekturbeschreibung.

Zusätzlich zu einer Klassifikation der Produkte sollen in diesem Arbeitspaket auch die bisher verwendeten Beschreibungstechniken katalogisiert und in Beziehung zu den Produktklassen gesetzt werden. Ausgehend von den bereits verwendeten Techniken, beispielsweise Komponentenbeschreibungen in Darwin [7], verfeinerten Strukturbeschreibungen in UML [6] und Abhängigkeitsbeschreibungen mittels Design Spaces [11], soll in dem Katalog die Relevanz einzelner Techniken für bestimmte Domänen dokumentiert werden. Die Kataloge können anschließend bei der Instanziierung des Produktmodells dazu benutzt werden, die für die jeweilige Zieldomäne erforderlichen Produkttypen sowie geeignete Beschreibungstechniken auszuwählen. Die Kataloge sollen in Zusammenarbeit mit den Teilprojekten A1 und A3 in die Erfahrungsdatenbank integriert werden.

Ein weiterer Untersuchungsschwerpunkt ist die Formulierung von Konsistenzbedingungen zwischen den verschiedenen Produkten; beispielsweise soll ausgedrückt werden können, dass bestimmte Elemente einer Struktur in weiteren Strukturen auftreten müssen, oder dass für bestimmte Strukturelemente jeweils ein anderes Dokument eines bestimmten Typs vorliegen muss.

Die Entwicklung des generischen Produktmodells muss verzahnt mit der Bereitstellung des generischen Prozessmodells im Arbeitspaket A2 erfolgen, da zwischen beiden Modellen enge wechselseitige Abhängigkeiten bestehen.

Personalbedarf: 10 PM

Arbeitspaket A2: Prozessmodell für die Anwendungsentwicklung mit vorkonfektionierten Systemen

Das Konzept vorkonfektionierter Systeme sieht vor, die Entwicklung von Anwendungen durch die Bereitstellung eines auf das spezifische System zugeschnittenen Entwicklungsprozesses zu unterstützen. Dazu soll in diesem Arbeitspaket ein generisches Prozessmodell für die Anwendungsentwicklung mit vorkonfektionierten Systemen erarbeitet werden, das die Grundlage für die Ableitung von maßgeschneiderten Entwicklungsprozessen für einzelne vorkonfektionierte Systeme bietet. Insbesondere ist dabei zu untersuchen, inwieweit das generische Prozessmodell offen gestaltet werden kann, um die Integration von für einzelne vorkonfektionierte Systeme spezifischen Aktivitäten und Entwicklungstechniken zu erlauben, beispielsweise von auf die Architektur des vorkonfektionierten Systems bzw. auf die wiederverwendbaren Artefakte zugeschnittenen Analyse- und Designmethoden.

Zur Entwicklung des generischen Prozessmodells soll zunächst die prinzipielle Vorgehensweise für Anwendungsentwicklung mit vorkonfektionierten Systemen beschrieben werden. Als Grundlage dienen die in der vergangenen Förderungsperiode begonnenen Beschreibungen generischer Entwicklungsaktivitäten, die die Integration wiederverwendbarer Strukturen und Produkte in verschiedenen Phasen des Entwick-

lungsprojektes aufzeigen [8]. Ausgehend von diesen Beschreibungen sollen die durchzuführenden Entwicklungsaktivitäten auf abstrakter, systemunabhängiger Ebene identifiziert werden. Anschließend sollen diese Entwicklungsaktivitäten in dem zu erstellenden generischen Prozessmodell beschrieben werden, für das eine weitgehende Kompatibilität zu dem in der vergangenen Förderungsperiode erstellten SFB-Referenzprozessmodell angestrebt wird. Bei der Prozessmodell-Realisierung soll darüberhinaus soweit wie möglich auf im SE-Labor bereits vorhandene Modellierungstechniken zurückgegriffen werden, beispielsweise auf im Teilprojekt B1 entwickelte Techniken zur Beschreibung von generischer Prozessmodellen. Das zu erstellende Prozessmodell muss die spezifischen Eigenschaften des in Arbeitspaket A1 entwickelten Produktmodells berücksichtigen.

Personalbedarf: 10 PM

Arbeitspaket A3: Werkzeugunterstützung für Anwendungsentwicklung mit vorkonfektionierten Systemen

In diesem Arbeitspaket soll auf Basis der in A1 und A2 bereitgestellten Produkt- und Prozessmodelle eine umfassende Werkzeugunterstützung für die Anwendungsentwicklung mit vorkonfektionierten Systemen realisiert werden. Dazu muss zunächst ein Werkzeugrahmen geschaffen werden, der die Grundlage und Kommunikationsplattform für speziellere Werkzeuge bietet. In diesen Rahmen soll ein geeignetes Prozessleitwerkzeug integriert werden, das einerseits eine uniforme Schnittstelle für den Zugriff auf die einzelnen Produkte des vorkonfektionierten Systems bietet, und andererseits Prozesswissen zugänglich macht, beispielsweise Abhängigkeiten zwischen Produkten oder Vorschläge für durchzuführende Aktivitäten. Das Prozessleitwerkzeug soll als Spezialisierung bzw. Erweiterung der in Teilprojekt A2 erstellten MILOS-Umgebung für Prozessmodellierung und -ausführung realisiert werden.

Mit dem Werkzeugrahmen werden darüberhinaus die Voraussetzungen für die Integration von Werkzeugen zur Darstellung verschiedener architektureller Sichten sowie zur Unterstützung spezifischer Methoden bzw. Entwicklungsaktivitäten geschaffen. Das Ziel ist hierbei, zu jedem vorkonfektionierten System einen darauf abgestimmten Satz an Werkzeugen zur Verfügung zu stellen, mit dem sowohl allgemeine Aspekte der Anwendungsentwicklung mit vorkonfektionierten Systemen als auch für das betrachtete System spezifische Aktivitäten unterstützt werden können.

Personalbedarf: 19 PM

Arbeitspaket A4: Validierung der erarbeiteten Modelle und Werkzeuge

In diesem Arbeitspaket sollen durch Fallstudien die Annahmen bezüglich der Flexibilität der generischen Produkt- und Prozessmodelle sowie bezüglich des mit vorkonfektionierten Systemen erzielbaren Wiederverwendungsgrades bestätigt werden. Dazu sollen zunächst die in der abgelaufenen Förderungsperiode realisierten vorkonfektionierten Systeme [15][16] für verteilte Gebäudeautomationssysteme mit Hilfe des in Arbeitspaket A1 erarbeiteten Produktmodell-Rahmens beschrieben werden.

Dabei soll gezeigt werden, dass insbesondere die hier verwendeten speziellen Dokumenttypen und Beschreibungstechniken in diesen Rahmen passen. Auf ähnliche Weise soll die Flexibilität und die Eignung des in Arbeitspaket A2 erstellten Prozessmodell-Rahmens nachgewiesen werden; hier ist zu bestätigen, dass auf Basis dieses Rahmens die für das betrachtete vorkonfektionierte System spezifischen Aktivitäten und Produkte und deren Verwendung problemlos modelliert werden können.

Desweiteren soll durch exemplarische Entwicklungsprojekte auf Basis eines in der vorangegangenen Förderungsperiode erstellten vorkonfektionierten Systems für Gebäudeautomation und des darauf zugeschnittenen Prozessmodells die Annahme bestätigt werden, dass durch das Wiederverwendungskonzept des Teilprojekts B10 eine Produktivitätssteigerung erzielt werden kann. Dazu ist insbesondere der Wiederverwendungsgrad zu messen, da im Konzept vorkonfektionierter Systeme umfassende Wiederverwendung den Schlüssel für eine Steigerung der Produktivität darstellt. Desweiteren sollen in diesen Entwicklungsprojekten die Standardmetriken Entwicklungszeit und Fehlerhäufigkeit zum Einsatz kommen, um die erzielte Produktivitätssteigerung zu quantifizieren.

Personalbedarf: 15 PM

Arbeitsschwerpunkt B: Verfahren zur Entwicklung und Wartung vorkonfektionierter Systeme

Arbeitspaket B1: Vorgehensmodell zur Entwicklung vorkonfektionierter Systeme

In diesem Arbeitspaket soll untersucht werden, wie mit dem Konzept vorkonfektionierter Systeme neue Domänen erschlossen werden können. Dazu muss ein Vorgehensmodell zur systematischen Entwicklung vorkonfektionierter Systeme erarbeitet werden. Grundlage ist die in der vergangenen Förderungsperiode in einem Querschnittsprojekt erarbeitete objektorientierte Domänenanalyse-Methode [13]. Es ist zu untersuchen, wie ein aus dieser Analyse resultierendes Domänenmodell systematisch in ein vorkonfektioniertes System umgesetzt werden kann. Bei der Realisierung einer passenden Systemarchitektur beispielsweise ist zu untersuchen, wie die im Domänenmodell festgehaltenen Gemeinsamkeiten und variablen Aspekte auf diese Architektur abgebildet werden können. Als Teil dieses Entwicklungsschrittes können beispielsweise mehrere Architekturvarianten entworfen und dann mit Hilfe der in Arbeitspaket C2 entwickelten Analysemethoden auf ihre Eignung für die Anforderungen der Domäne untersucht werden.

Wesentlicher Bestandteil der Entwicklung eines neuen vorkonfektionierten Systems ist die Instanziierung der in Arbeitspaket A1 und A2 entwickelten generischen Produkt- bzw. Prozessmodelle im Kontext der betrachteten Domäne. In diesem Arbeitspaket soll daher weiterhin eine Vorgehensweise für deren Instanziierung bereitgestellt werden, die es erlaubt, einen gezielt auf das jeweilige vorkonfektionierte System zugeschnittenen Entwicklungsprozess zu realisieren. Dies soll im wesentlichen durch Spezialisierung der generischen Entwicklungsaktivitäten erreicht werden, unter ande-

rem durch Identifikation der Ausgangsprodukte des jeweiligen Entwicklungsschritts sowie der zu erstellenden bzw. zu modifizierenden Produkte.

Personalbedarf: 12 PM

Arbeitspaket B2: Wartung vorkonfektionierter Systeme

In diesem Arbeitspaket sollen Konzepte für die systematische Durchführung von Änderungen an vorkonfektionierten Systemen erarbeitet werden. Dazu werden zunächst in mehreren Fallstudien charakteristische Änderungsszenarien an Hand des erwähnten vorkonfektionierten Systems für Gebäudeautomation betrachtet, die sich vor allem in Bezug auf die Tiefe der Eingriffe in das System unterscheiden. In vielen Fällen können die Änderungsanforderungen durch Wartung an einzelnen Elementen abgehandelt werden, in anderen Fällen ist eine strukturelle Modifikation erforderlich. Im schlimmsten Fall sind Änderungen am unterliegenden Domänenmodell notwendig, die unter Umständen weitreichende Folgen für den gesamten Systementwurf haben.

Die Durchführung von Änderungen stellt zusätzliche Anforderungen an die Dokumentation eines vorkonfektionierten Systems. An Hand der Fallstudien soll daher ermittelt werden, inwieweit die bisher definierte Dokumentation ausreicht, bzw. an welchen Stellen das Produktmodell noch erweitert werden muss, um Änderungsanforderungen in das vorkonfektionierte System einzuarbeiten. Des Weiteren soll auf Basis der Erfahrungen aus den Fallstudien ein Vorgehensmodell entwickelt werden, das spezifisch für die jeweilige Tiefe des Eingriffes die Vorgehensweise zur Änderung des vorkonfektionierten Systems beschreibt.

Ein weiterer Untersuchungsschwerpunkt liegt auf der Überwachung der Konsistenz des vorkonfektionierten Systems: die vorgenommenen Änderungen müssen mit den existierenden Systembestandteilen und den ihnen zugrundeliegenden Strategien und Mechanismen verträglich sein. Um dies sicherzustellen, kann zum einen das vorliegende architekturelle Wissen herangezogen werden, etwa die angewendeten prinzipiellen Lösungsstrategien oder die Beschreibung der Interaktionsmuster der Systemkomponenten. Zum anderen können auch die betreffenden Design Spaces genutzt werden, um durch die Auswertung von Korrelationen beispielsweise zwischen Anforderungen und Lösungsstrukturen potentielle Inkonsistenzen aufzudecken. Auf diesem Problemgebiet soll auch in Zusammenarbeit mit dem Teilprojekt C1 nach Möglichkeiten für eine formale Beschreibung von Konsistenzbedingungen und darauf basierende automatisierte Konsistenzüberprüfungen gesucht werden.

Personalbedarf: 12 PM

Arbeitspaket B3: Werkzeugunterstützung für Entwicklung und Wartung vorkonfektionierter Systeme

Auf Basis der konzeptionellen Arbeiten über die Entwicklung und Wartung vorkonfektionierter Systeme in den Arbeitspaketen B1 und B2 sollen in diesem Arbeitspaket

geeignete Unterstützungswerkzeuge realisiert werden. Die Unterstützung der Entwicklung vorkonfektionierter Systeme zielt dabei schwerpunktmäßig auf die frühen Phasen der Systementwicklung ab. Insbesondere soll hier ein Werkzeug für die Durchführung der objektorientierten Domänenanalyse entwickelt werden, sowie Unterstützung für die anschließende Dokumentation von Zusammenhängen zwischen Alternativen im Domänenmodell und Abstraktionen bzw. spezifischen wiederverwendbaren Produkten im vorkonfektionierten System. Als Umgebung für die Instanziierung des generischen Prozessmodells für einzelne vorkonfektionierte Systeme soll eine spezialisierte Variante des in der vergangenen Förderungsperiode im Teilprojekt A2 erstellten MILOS-Werkzeugs genutzt werden.

Zur Unterstützung der Wartung vorkonfektionierter Systeme sollen Werkzeuge entwickelt werden, die die Identifikation der von Änderungsanforderungen betroffenen Systembestandteile sowie die Abschätzung der Auswirkungen von Änderungsanforderungen erlauben. Dazu kann zum einen die sogenannte Uses-Struktur des Systems genutzt werden, die Abhängigkeitsbeziehungen zwischen verschiedenen Strukturelementen dokumentiert, zum anderen der zum vorkonfektionierten System gehörende Design Space, der Zusammenhänge zwischen Anforderungen und Systemkomponenten aufzeigt. Darüberhinaus können die Korrelationen im Design Space genutzt werden, um bei der Durchführung von Änderungen die Konsistenz des Systems zu überwachen.

Personalbedarf: 21 PM

Arbeitsschwerpunkt C: Mess- und Analysetechniken

Arbeitspaket C1: Mess- und Analysetechniken für die Anwendungsentwicklung mit vorkonfektionierten Systemen

In diesem Arbeitspaket sollen zur Unterstützung der Anwendungsentwicklung mit vorkonfektionierten System geeignete Mess- und Analysetechniken ausgearbeitet werden. Als Basistechnik soll dabei die Architekturanalyse nach der in der vergangenen Förderungsperiode entwickelten erweiterten SAAM-Methode [3][11] eingesetzt werden.

Neben der Integration von Mess- und Analyseverfahren für Validierungszwecke sollen Techniken entwickelt werden, um gezielt die Durchführung von Wiederverwendungsaktivitäten zu unterstützen. Dazu können beispielsweise geeignete Szenarien in vorkonfektionierte Systeme aufgenommen werden, die grundlegende Eigenschaften seiner Architektur illustrieren. Anhand dieser Szenarien kann abgeschätzt werden, inwieweit sich die verfügbaren wiederverwendbaren Artefakte für einen speziellen Anwendungsfall eignen, bzw. wie groß der erwartete Aufwand für die Integration der Artefakte in das fertige System ist. Desweiteren sollen Mess- und Analysetechniken zur Integration in vorkonfektionierte Systeme untersucht werden, die auf die Umsetzung spezifischer, durch das System unterstützter nichtfunktionaler Eigenschaften hin ausgelegt sind. Für die Gestaltung des Zeitverhaltens von Anwendungen beispiels-

weise kann die SAAM-Methode mit dafür geeigneten ebenfalls szenariobasierten Analysemethoden kombiniert werden.

Personalbedarf: 6 PM

Arbeitspaket C2: Mess- und Analysetechniken für die Entwicklung vorkonfektionierter Systeme

In diesem Arbeitspaket sollen geeignete Mess- und Analysetechniken auch für die Neuentwicklung vorkonfektionierter Systeme untersucht werden, wiederum basierend auf der in der vergangenen Förderungsperiode entwickelten erweiterten SAAM-Methode. Zum einen soll hier Architekturanalyse für Validierungszwecke und zur Analyse der architekturellen Komplexität eingesetzt werden. Zum anderen sollen Mess- und Analysetechniken zur Beurteilung der Flexibilität vorkonfektionierter Systeme untersucht werden. Aufgabe dieser Analyse ist einerseits der frühzeitige Vergleich mehrerer Systemdesign-Varianten mit dem Ziel, das Design zu finden, das die in der Domäne geforderte Flexibilität am besten umsetzt. Die diesbezüglichen Analyseergebnisse stellen weiterhin einen wesentlichen Beitrag zur Dokumentation des vorkonfektionierten Systems dar, da sie seinen Einsatzbereich präzise umreißen. Zusammen mit den zur Analyse benutzten Szenarien können die Analyseergebnisse auch dazu dienen, die Eignung eines vorkonfektionierten Systems für eine bestimmte Anwendung bzw. Alternativen bezüglich verschiedener vorkonfektionierter Systeme zu bewerten.

Eine weitere Zielsetzung für die Untersuchung von Mess- und Analysetechniken liegt darin, in Entwicklung befindliche vorkonfektionierte Systeme im Hinblick auf spezifische nichtfunktionale Eigenschaften zu evaluieren. Dies dient einerseits der Validierung, d.h. der Überprüfung, ob das Systemdesign bestimmte geforderte Eigenschaften erbringt; andererseits liefern auch diese Analyseergebnisse sowie die dafür benutzten Szenarien wieder wesentliche Beiträge zur Dokumentation des vorkonfektionierten Systems und können bei der Entscheidung über seinen Einsatz herangezogen werden.

Personalbedarf: 6 PM

Arbeitspaket C3: Werkzeugunterstützung für Mess- und Analysetechniken

Zur Unterstützung der Anwendung der in den Arbeitspaketen C1 und C2 entwickelten Mess- und Analyseverfahren sollen in diesem Arbeitspaket geeignete Werkzeuge realisiert werden. So soll zunächst die Durchführung von Architekturanalysen nach der erweiterten SAAM-Methode unterstützt werden, die die Basistechnologie für die spezielleren Analyseverfahren darstellt. Eine spezielle Anforderung ist die Unterstützung einer wiederholten und vergleichenden Analyse derselben Menge von Szenarien in verschiedenen Entwicklungsstadien eines vorkonfektionierten Systems, beispielsweise während der Entwicklung der Architektur und nach Erstellung der Komponentenentwürfe. Dazu ist das Aufzeichnen von Analyse-Historien für spätere Vergleiche notwendig, die das Verfolgen bestimmter Entwurfsgesichtspunkte durch die verschiedenen Entwicklungsaktivitäten ermöglichen; insbesondere wird dadurch eine Brücke

von den Elementen und Mechanismen des vorkonfektionierten Systems zum Design der einzelnen Anwendung geschlagen.

Das zu realisierende Analyse-Basiswerkzeug soll in den im Arbeitsschwerpunkt A erstellten Werkzeugrahmen eingepasst werden; insbesondere muss es Schnittstellen zu den in den Arbeitspaketen A3 und B3 erstellten Werkzeugen für die Anwendungsentwicklung mit vorkonfektionierten Systemen bzw. die Neuentwicklung solcher Systeme zur Verfügung stellen. Darauf aufbauend können Spezialwerkzeuge für besondere Analysezwecke konstruiert werden, beispielsweise für die in Arbeitspaket C1 skizzierte szenariobasierte Analyse zeitkritischer Abläufe in Anwendungen.

Personalbedarf: 15 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausstattung						
A. Gilbert	C1			C2		
	C3					
Ergänzungsausstattung						
G. Molter	A1 und A2			A3		
	A4					
L. Geyer	B1			B2		
	B3					
Stud. Hilfskräfte:						
N.N.	A3 und A4					
N.N.	B3 und C3					

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

B10 bildet mit den Teilprojekten B4, B5 und A3 diejenige Gruppe von Teilprojekten, die an generischen Produktmodellen arbeiten. Deshalb gibt es unter diesen Teilprojekten einen ständigen Erfahrungsaustausch, insbesondere im Bereich von Produkt-Modellierungstechniken. Darüberhinaus soll der in Arbeitspaket A1 erstellte Katalog von Produkttypen und Entwicklungstechniken in Zusammenarbeit mit den Teilprojekten A1 und A3 in die Erfahrungsdatenbank eingebracht werden. Ein enger Erfahrungsaustausch ist auch mit dem Teilprojekt B1 vorgesehen, auf dessen Ansätze zur generischen Prozessmodellierung bei der Erstellung des Prozessmodells für die Anwendungsentwicklung mit vorkonfektionierten Systemen zurückgegriffen werden soll. Ähnliches gilt für das Teilprojekt A2, dessen MILOS-Umgebung für Prozessmodellierung und -ausführung zur Realisierung des Prozessleitwerkzeuges spezialisiert bzw. erweitert werden soll.

Bei der angestrebten Formalisierung des Design Space-Konzeptes, mit dem wir Zusammenhänge zwischen Anforderungen und Lösungsalternativen modellieren, haben wir eine enge Kooperation mit dem Teilprojekten C1, D1 und D2 begonnen.

Die Teilprojekte B10, B12 und D1 bilden eine Gruppe von Teilprojekten, die sich mit Ansätzen zur wiederverwendungsorientierten Entwicklung von Softwaresystemen beschäftigen. Im Laufe der nächsten Förderungsperiode ist hier eine Reihe von Experimenten geplant, um diese alternativen Ansätze zu vergleichen.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	1	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.8.200
			StdHkr.	2	40.800	StdHkr.	2	40.800	StdHkr.	2	40.800
	zus.:	1	zus.:	4	249.600	zus.:	4	249.600	zus.:	4	249.600
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		3.000	(522)		3.000	(522)		3.000
			zusammen		3.000	zusammen		3.000	zusammen		3.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausstattung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Nehmer, J.; Prof. Dr. (2) Gilbert, A., Dipl-Inform.	Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL	5 20	1.1.95 15.1.00	
Ergänzungsausstattung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(3) Geyer, L., Dipl-Inform. (4) Molter, G., Dipl. Inform. (5) N.N., StdHkr. (6) N.N., StdHkr.	Prakt. Inf. Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	38,5 38,5 19 19	1.9.97 1.1.96 1.1.01 1.1.01	IIa IIa
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Der wissenschaftliche Mitarbeiter aus der Grundausrüstung ist für die Bearbeitung des Arbeitsschwerpunkts C vorgesehen.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Der erste wissenschaftliche Mitarbeiter ist für die Bearbeitung des Arbeitsschwerpunkts B vorgesehen.

Der zweite wissenschaftliche Mitarbeiter ist für die Bearbeitung des Arbeitsschwerpunkts A vorgesehen.

Die erste studentische Hilfskraft hilft bei Implementierungsarbeiten im Arbeitspaket A3 und bei der Validierung in Arbeitspaket A4.

Die zweite studentische Hilfskraft hilft bei Implementierungsarbeiten in den Arbeitspaketen B3 und C3.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausrüstung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausrüstung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausrüstung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausrüstung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
	-	-	-
Summe	-	-	-

3.1 Allgemeine Angaben zum Teilprojekt B11

3.1.1 Thema

Zuverlässige Eingebettete Systeme

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Entwurfsmethodik Eingebetteter Systeme

3.1.3 Leiter

PD Dr. Bernd Schürmann
geb. 22.08.1959
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-3069
Fax: 0631/205-2162
schuerma@informatik.uni-kl.de

Die Stelle des Leiters ist befristet: () nein (X) ja, bis 2007

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	() ja	(x) nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	() ja	(x) nein
Tierversuche	() ja	(x) nein
gentechnologische Untersuchungen	() ja	(x) nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Das Teilprojekt wird neu beantragt.

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
2001	124,8	3,0	25,0	152,8
2002	124,8	3,0	-	127,8
2003	124,8	3,0	-	127,8

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Die Komplexität der vom SFB betrachteten eingebetteten Systeme wird erheblich von deren nicht-funktionalen Anforderungen beeinflusst, weshalb diese im engeren Interesse des SFB liegen. Die wichtigsten nicht-funktionalen Anforderungen an eingebettete Systeme sind neben den Kosten die Echtzeit sowie die Zuverlässigkeit und Sicherheit.

In den ersten beiden Förderungsperioden wurden in diesem Zusammenhang Zeitaspekte vorrangig betrachtet. Der Aspekt Kosten wurde implizit durch die Untersuchung kostengünstiger Entwicklungsprozesse eingebracht. Mit diesem (neu beantragten) Teilprojekt sollen durch Betrachtung der Zuverlässigkeit und Sicherheit die nicht-funktionalen Aspekte eine noch stärkere Beachtung im SFB erlangen.

Man unterscheidet heute drei Ursachen für das Versagen eines eingebetteten Systems: Konstruktionsfehler, Ausfall einer Komponente und Bedienfehler. Zur Erhöhung der Zuverlässigkeit¹ eines solchen Systems kann man einerseits versuchen durch geeignete Maßnahmen im Entwurfsprozess (*Konstruktions-*) Fehler zu verhindern und andererseits *fehlertolerante* bzw. *robuste Systeme*, die alle drei Fehlerursachen tolerieren, entwickeln. Während andere Teilprojekte des SFB, Fortschritte bei der Fehlerverhinderung durch Qualitätssteigerung, u.a. aufgrund des SFB-spezifischen Zyklus „Messen, Bewerten, Verbessern“, erreichen wollen, soll der Schwerpunkt dieses Teilprojekts in der Entwicklung fehlertoleranter, robuster Systeme liegen.

Langfristiges Ziel des Teilprojekts ist die Erhöhung der Verfügbarkeit der entworfenen Anwendungssysteme, u.a. durch schrittweise Ersetzung von Systemfunktionalitäten durch Notlaufeigenschaften. Die Betrachtung der Zuverlässigkeit hat Auswirkungen auf alle Phasen des Entwicklungsprozesses und bildet einen Schwerpunkt in der Design-Phase. Unsere Vision ist es, aus einer im Wesentlichen funktionalen Spezifikation und einer gegebenen Zielplattform möglichst einfach (weitgehend automatisch) ein robustes System mit mehreren Ebenen von Notlaufeigenschaften zu entwerfen.

In diesem Teilprojekt soll dazu untersucht werden, wie bestehende Ansätze zur Entwicklung robuster, zuverlässiger Systeme (Modellierung und Bewertung von Zuverlässigkeit, Erhöhung der Zuverlässigkeit durch Redundanzbildung, Einführung von Notlaufeigenschaften etc.) in den Systementwicklungsprozess einfließen können. Drei wichtige Zuverlässigkeitsaspekte stehen dabei im Mittelpunkt der Betrachtungen:

- Bewertung der Funktionsfähigkeit und Sicherheit entworfenen Systeme
- Fehlertoleranz bzw. Robustheit zur Erhöhung der Funktionssicherheit
- Erhöhung der Zuverlässigkeit und Sicherheit durch Notlaufeigenschaften.

Existierende Ansätze und Verfahren im Umfeld der Zuverlässigkeitsbetrachtung sol-

1. Eine genauere Definition der Begriffe „Zuverlässigkeit“, „Robustheit“, „Sicherheit“ etc., wie sie in diesem Antrag verwendet werden, findet sich im folgenden Abschnitt.

len schrittweise in den Entwicklungsprozess des SFBs einfließen und so eine stufenweise Erhöhung der Robustheit und Zuverlässigkeit der im SFB entwickelten Systeme ermöglichen. Sehr interessant erscheinen in diesem Umfeld Fragestellungen bezüglich der gemeinsamen Betrachtung von Hardware und Software (während der Design-Phase) sowie generischer Produkte und Prozesse.

3.3 Stand der Forschung

Dieses Teilprojekt will sich mit der Entwicklung robuster, zuverlässiger eingebetteter Systeme beschäftigen. Es sollen existierende Verfahren in den SFB-Entwicklungsprozess aufgenommen werden. Daher müssen einerseits die Vorarbeiten auf dem Gebiet der Erhöhung von Zuverlässigkeit und Fehlertoleranz eingebetteter Systeme und andererseits Forschungsergebnisse zum systematischen Entwurf eingebetteter Systeme betrachtet werden.

Entwurfsmethodik eingebetteter Systeme

Die Untersuchung geeigneter Entwurfsprozesse für eingebettete Systeme ist eine zentrale Fragestellung mehrerer Teilprojekte. Die bisher im SFB erzielten Ergebnisse, vor allem im Bereich der Anforderungsanalyse, sind daher im Rahmen des „Standes der Forschung“ zu nennen. Sie sind im Ergebnisbericht nachzulesen. Diese Arbeiten sollen mit B11 weitergeführt werden.

Die Zuverlässigkeit als nicht-funktionale Eigenschaften muss schwerpunktmäßig in der Design-Phase betrachtet werden. Hier sind die Zielplattform und damit verstärkt Hardware- und Verteilungsaspekte mit zu berücksichtigen. Neben den Arbeiten zum Entwurf eingebetteter Software, die im Mittelpunkt der Arbeiten im SFB in den ersten beiden Förderungsperioden standen, werden nun auch hardwarenahe Aspekte interessant. Zum bisher betrachteten Stand der Technik im Bereich des Systementwurfs sind nun auch Arbeiten zum Hardware/Software-Codesign (verteilter Systeme), die sich aus der High-Level-Synthese entwickelt haben, hinzuzufügen.

Kern des Hardware/Software-Codesign ist eine geeignete Partitionierung des Gesamtsystems in Hardware und Software, woran sich dann der Komponentenentwurf für Hardware und Software getrennt anschließt. Wichtige Schritte zur Bewertung der Partitionierung sind die Hardware/Software-Cosimulation und darauf aufbauende, weitergehende Analysen. Beispiele für Hardware/Software-Entwurfsumgebungen, die den Stand der Technik repräsentieren, sind das POLIS Codesign Framework für eingebettete Systeme aus Berkeley [2] und Cosyma aus Braunschweig [31]. Während einzelne Techniken für unser Vorhaben übernommen werden können, unterscheidet sich unsere Zielsetzung von meisten Codesign-Ansätzen durch die zugrunde liegende Zielplattform. In vielen Fällen findet man einen Koprozessor-Ansatz, bei dem zeitkritische Routinen in noch zu synthetisierender Hardware ausgeführt werden, wogegen wir eine verteilte Zielplattform bestehend aus vorhandenen Komponenten konfigurieren wollen.

In dieses Umfeld des Hardware/Software-Codesign sind auch die jüngeren Bestrebungen einzuordnen, aus Verhaltensspezifikationen - modelliert in Analysesprachen wie StateMate und SDL - automatisch VHDL-Code und damit Hardware zu synthetisieren (z.B. [7]). Dies ermöglicht eine einheitliche Modellierungs- und Entwurfsumgebung für Hardware und Software.

Der Entwurf größerer eingebetteter Systeme ist heute (noch) nicht vollautomatisch denkbar. Hier müssen die Zielplattform und die darauf laufende eingebettete Software von mehr oder weniger großen Entwurfsteams aus unterschiedlichen Disziplinen in vielen Teilschritten entworfen werden. Zur Lösung dieser Aufgabe sind weniger die Algorithmen der einzelnen Schritte als vielmehr die gesamte Entwurfsmethodik in den Mittelpunkt des Interesses zu rücken. Aufgrund des großen Interesses am Entwurf solcher größerer eingebetteten Systeme findet man auch hier bereits eine Vielzahl von Arbeiten. Stellvertretend für diese sollen das DFG-Schwerpunktprogramm „Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen“ [11] und der Sonderforschungsbereich 358 „Automatisierter Systementwurf“ [35] genannt werden. Auch das DFG-Schwerpunktprogramm „Entwurf und Entwurfsmethodik eingebetteter Systeme“ [10] hatte viele Berührungspunkte zu unseren Fragestellungen.

Der DFG-Schwerpunkt „Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen“ beschäftigt sich mit der Integration von Modellen aus der Informatik und den Ingenieurwissenschaften. Dies ist besonders wichtig, wenn die eingebettete Software auf ein Zielsystem angepasst werden soll. Die Sicherstellung der Verfolgbarkeit durch integrierte Modelle kann die Zuverlässigkeit eingebetteter Systeme wesentlich erhöhen. Solche integrierten Modelle findet man heute allerdings erst vereinzelt in speziellen Anwendungsbereichen.

Der Sonderforschungsbereich 358 „Automatisierter Systementwurf“ in Dresden verfolgt einen ähnlichen Design-Ansatz wie wir. Ein Projektbereich versucht mit Hilfe formalisierter Methoden spezielle Klassen von Algorithmen auf ausgewählte Rechnerarchitekturen abzubilden. Der Schwerpunkt der Codesign-Arbeiten liegt in der Untersuchung der wechselseitigen Beeinflussung von Algorithmen und Architektur, um zu einer effizienten Algorithmenabbildung durch ein automatisiertes Entwurfssystem zu gelangen. Unterschiedlich zu uns ist das Anwendungsfeld und die damit zusammenhängende Zielarchitektur (eng gekoppelte Rechenfelder für rechenintensive Algorithmen). Ein weiterer Projektbereich beschäftigt sich mit Systemarchitekturen für verteilte Echtzeitsysteme und der dazu notwendigen Betriebsmittelplanung auf allen Systemebenen. Darüber hinaus liegt auch in der Zielsetzung, wiederverwendbare, generische Produkte zu entwickeln, ein gemeinsamer Forschungsaspekt. Bezüglich nicht-funktionaler Eigenschaften, speziell der uns interessierenden Zuverlässigkeit, liegt der Arbeitsschwerpunkt in Dresden auf dem Testen und der Simulation analog/digitaler Schaltungen.

Betrachtet man durchgängige Entwurfsmethodiken für eingebettete Systeme, so werden nicht nur in den genannten Projekten vorrangig Zeitaspekte untersucht. Beispiels-

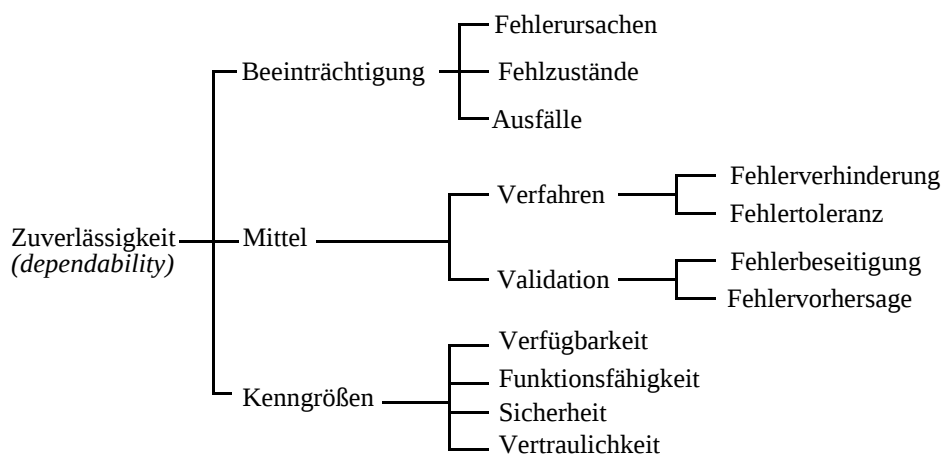
weise arbeitet auch die Forschungsgruppe von Prof. Nett aus Magdeburg gemeinsam mit der GMD und anderen Partnern an einer integrierten Entwurfsumgebung für Echtzeitsysteme [4]. Als weiteres Beispiel sei das in München in der Forschungsgruppe von Prof. Färber entwickelte REAR-Framework genannt, bei dem die Entwicklung von Echtzeitsystemen auf annotierten SDL-Modellen basiert [32]. Eine vollständige Liste von Arbeiten auf dem Gebiet der Entwicklung von Echtzeitsystemen würde allerdings den Umfang dieses Antrags sprengen.

Alle diese Arbeiten zum Entwurf eingebetteter Systeme unter Beachtung von Zeitbedingungen sehen wir als gegebene Vorarbeiten, auf denen wir mit der Betrachtung der Zuverlässigkeit aufbauen wollen.

Zuverlässigkeit und Robustheit eingebetteter Systeme

Sicherheit, Zuverlässigkeit und Robustheit¹ sind wesentliche Merkmale eingebetteter Systeme. Sie betreffen sowohl das technische System selbst - einschließlich der mechanischen und elektronischen Komponenten - als auch die eingebettete Software, die in der Regel das technische System überwacht und steuert, wobei der Mensch als Bediener nicht unberücksichtigt bleiben darf. Vorarbeiten auf dem Gebiet des Entwurfs zuverlässiger Systeme findet man daher zum einen in den Ingenieurfächern, die sich mit den technischen Systemen beschäftigen, und zum anderen im Umfeld der Automatisierungstechnik und des Software-Engineering. Auch wenn die Betrachtung der Zuverlässigkeit stark praxisorientiert ist und die verschiedenen Ingenieurdisziplinen ihre eigenen Begriffe und Lösungen verwenden, gibt es bereits einen großen

1. Begriffe wie „Sicherheit“ und „Zuverlässigkeit“ werden häufig unterschiedlich verwendet. Wir wollen in diesem Antrag den Definitionen nach [29] folgen:



Solange im Text nicht zwischen diesen Begriffen differenziert werden muss, wird der Einfachheit halber im Folgenden nur noch von „Zuverlässigkeit“ gesprochen. Wir folgen in diesem Antrag den Bezeichnungen aus [29], auch wenn in jüngerer Zeit im Deutschen häufig für den allgemeinen Begriff „dependability“ der Begriff „Verlässlichkeit“ verwendet wird. „Zuverlässigkeit“ tritt dann anstelle des Begriffs „Funktionsfähigkeit“ als Übersetzung für „reliability“.

Unter einem robusten, fehlertoleranten eingebetteten System soll demnach ein System verstanden werden, das trotz Vorliegen von Fehlern (Fehlerursachen) nicht in einen Fehlzustand gerät und auch nicht ausfällt.

Katalog von Normen und Richtlinien bezüglich der Zuverlässigkeit eingebetteter Systeme (vgl. u.a. [13], [30]), die nicht außer Acht gelassen werden dürfen.

Die Grundlage für die Entwicklung sicherheitsgerichteter Rechnersysteme bildet die DIN V VDE 0801 [14] („Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben“). Diese Norm baut auf DIN 19250 [13] („Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen“) auf. Basis für diese Normen sind die DIN VDI/VDE 3542 [15] zu den sicherheitstechnischen Begriffen für Automatisierungssysteme und die DIN 31000 Teil 2 zu allgemeine Leitsätze der Sicherheitstechnik. Letztere wurde inzwischen durch DIN EN 292 [18] („Sicherheit von Maschinen“) und DIN VDE 1000 [17] („Allgemeine Leitsätze für das sicherheitsgerechte Gestalten technischer Erzeugnisse“) ersetzt. Die Begriffe der Zuverlässigkeit sind in DIN 40041 [16] festgelegt.

Bei der Entwicklung zuverlässiger eingebetteter Systeme sind diese Vorschriften und die geforderten Maßnahmen zu berücksichtigen und einzuhalten. In der industriellen Praxis sind heute in den meisten Fällen qualitative und quantitative Nachweise über die Einhaltung dieser Richtlinien zu führen.

Lösungsansätze zur Erhöhung der Zuverlässigkeit bzw. zum Erreichen eines gewünschten Grads an Zuverlässigkeit sind einerseits Arbeiten im Bereich Fehlerverhinderung und Qualitätssicherung und andererseits Arbeiten auf dem Gebiet der Fehlertoleranz. Letztere werden notwendig, da (Hardware-) Ausfälle von Komponenten des technischen Systems und Bedienfehler in Betracht gezogen werden müssen und heutzutage die Korrektheit eines Entwurfs i. allg. noch nicht vollständig verifiziert werden kann. Fehlerverhinderung und Fehlertoleranz verfolgen zwar grundsätzlich unterschiedliche Ziele, enthalten aber mit der Fehlermodellierung gemeinsame Teilaspekte.

Fehlerverhinderung, Qualitätssicherung

Die wichtigsten Aktionen zur Fehlerverhinderung und Qualitätssicherung sind die Verifikation, die Validierung, das Testen und die Simulation anhand eines Modells oder eines Prototyps. Die Verifikation dient dem Nachweis, dass das realisierte System der Spezifikation entspricht. Sie findet in allen Phasen der Systementwicklung statt. Hilfsmittel zur Verifikation reichen von Inspektionsmethoden bis zu formalen Ansätzen, worunter auch Generatoren fallen können („correctness by construction“). Durch vermehrte Verwendung von Generatoren, Transformatoren und Bibliotheken in den späteren Entwurfsphasen erlangt die Verifikation der Analysephase immer mehr an Bedeutung. Die hierfür existierenden Methoden lassen sich in vorwärtsgerichtete induktive Methoden (z.B. Event Tree Analysis, Failure Mode and Effect Analysis) und rückwärtsgerichtete deduktive Methoden (z.B. Fault Tree Analysis, Probabilistic Safety Assessment) unterteilen [36]. Voraussetzung zur Automatisierung solcher Methoden ist in der Regel eine formale Spezifikation, die allerdings häufig nicht gegeben ist. Wichtige Beiträge auf dem Gebiet der formalen Verifikation leisteten in jüngerer Zeit die vom BMBF geförderten Forschungsprojekte „KorSo/KorSys“ [6] und „Espress“ [20]. Im Bereich der Hardware wurde bereits der Entwurf eines Prozes-

sors mit formalen Methoden als korrekt bewiesen [22]. Bei informellen Spezifikationen sind diese Verifikationsmethoden manuell auszuführen. Diese werden dann um Inspektionsmethoden ergänzt, wie sie u.a. im Teilprojekt B1 und im Fraunhofer-Institut für Experimentelles Software-Engineering (IESE) vor allem im Bereich der eingebetteten Software untersucht werden [28].

Rapid Prototyping spielt neben dem Testen eine immer wichtiger werdende Rolle zur Validierung eingebetteter Systeme. Während das Testen der Übereinstimmung der Systemspezifikation mit der Problembeschreibung durch den Systementwickler „lediglich“ das Ergebnis der Analysephase verifizieren kann, hilft das Prototyping auch die Kundenanforderungen einer Überprüfung zu unterziehen. Hilfsmittel hierfür sind die Simulatoren und Codegeneratoren von Analysewerkzeugen, wie beispielsweise Statemate und SDT, aber auch schnelle Prototypumgebungen für Hardwaresysteme basierend auf programmierbaren Schaltungen. In jüngerer Zeit beschäftigt sich ein eigener DFG-Schwerpunkt mit diesen Themen des Rapid Prototyping [12].

Fehlertoleranz, Robustheit

Die Fehlerverhinderung und Qualitätssicherung eingebetteter Systeme durch Validierung, Test und Verifikation, aber auch mittels verbesserter Entwicklungsprozesse, als eine zentrale Fragestellung aller Teilprojekte des SFB ist nicht das zentrale Thema dieses Teilprojekts. Hier soll vielmehr die Entwicklung robuster, fehlertoleranter Systeme untersucht werden, auch wenn dies nicht ganz losgelöst von der Fragestellung der Fehlerverhinderung und Qualitätssicherung gesehen werden kann. Das Teilprojekt sieht seine Aufgabe in der Untersuchung, wie bestehende Ansätze zur Fehlermodellierung und Fehlertoleranz in den Systementwicklungszyklus übernommen werden können. Ein wesentlicher Aspekt ist die Betrachtung von Notlaufeigenschaften.

Den Stand der Technik in Bezug auf diese zentralen Fragestellungen findet man bei den Arbeiten zu den verschiedenen Redundanzansätzen, zur Fehlermodellierung bzw. Zuverlässigkeitsanalyse und zu Entwurfsprozessen zuverlässiger und sicherer Systeme.

Im Bereich der *Zuverlässigkeitsmodellierung* eingebetteter Systeme wurden bereits viele Vorarbeiten geleistet und Ergebnisse erzielt. Die ersten Arbeiten reichen bereits über dreißig Jahre zurück (z.B. [3], [24]). Alle wesentlichen Arbeiten in Bezug auf Fehlerwahrscheinlichkeiten basieren auf stochastischen Modellen (häufig Markoffsche Prozesse), womit statistische Aussagen über Betriebs- und Ausfallzustände gemacht werden können. Die mathematischen Grundlagen bilden heute bereits ein etabliertes Fachgebiet und sind in vielen Grundlagenbüchern enthalten (z.B. in [22]). Viele der Modellierungen sind recht genaue Näherungslösungen, jedoch nur für kleinere Systeme geeignet. Jüngere Arbeiten beschäftigen sich daher vermehrt mit der Analyse der Zuverlässigkeit großer, komplexer und/oder verteilter Systeme (z.B. [27]). Gemeinsam ist allen stochastischen Modellen die Eigenschaft, dass sie Ausfallwahrscheinlichkeiten sehr gut modellieren und damit helfen, Zuverlässigkeitsaussagen zu treffen.

Für die Fehlersimulation, egal, ob mit Hilfe eines Simulators oder eines Prototyps, und für Robustheitsanalysen genügen in der Regel einfachere Fehlermodelle wie beispielsweise Aufzählungslisten oder Fehlerbäume [5]. Durch solche Modelle lässt sich untersuchen, welche Auswirkungen einzelne Fehler auf das Gesamtsystem haben. Dies kann dann Grundlage der Festlegung von Notlaufeigenschaften sein.

Die Fehler selbst können nur an den Schnittstellen zu fehlerfreien Objekten erkannt werden, was bei der Systempartitionierung in der Design-Phase beachtet werden muss. Frühere Arbeiten haben bereits gezeigt, dass hierarchische Strukturen den Umfang notwendiger Fehlermodelle drastisch reduzieren können [5]. Fehlertoleranzinstanzen müssen nur lokale Fehler tolerieren.

Ein heutzutage weit verbreiteter Ansatz ist FME(C)A (Failure Modes Effects and Criticality Analysis [8]). Es handelt sich hierbei um ein Werkzeug zur Identifikation, Analyse und Priorisierung potentieller Fehler. Ausgehend von sehr frühen Arbeiten Ende der 50er Jahre ist FME(C)A heute integraler Bestandteil aller QS 9000¹-Qualitätssysteme. Man findet diesen Ansatz bereits in einer Reihe kommerzieller Systeme.

Die Grenze zwischen *Hardware- und Software-Redundanz* ist bei eingebetteten Hardware/Software-Systemen fließend. Bei den Hardware-Ansätzen unterscheidet man zwischen statischer, dynamischer oder hybrider Redundanz [23]. Die statischen Verfahren beruhen auf der Maskierung von Fehlern. Die Komponenten werden dupliziert, und erfüllen parallel und synchron zueinander die gleiche Aufgabe. Ein ‘Voter’ entscheidet anschließend durch Vergleich der Ergebnisse, ob ein Fehler aufgetreten ist bzw. welches das korrekte Ergebnis ist. Beispiele hierfür sind die klassischen m-von-n-Systeme ($m \leq n$), wie z.B. die 2-von-2 Systeme zur Fehlererkennung und 2-von-3-Systeme zur Fehlermaskierung, die in der ADES (Architecture for Dependable Embedded Systems) [19] zur Realisierung des ‘Fail-safe’- bzw. ‘Fail-operational’-Modus eingesetzt werden.

Die dynamischen Redundanzansätze bauen auf Fehlererkennung, Fehlerlokalisierung und der Rekonfiguration, d.h. dem Ersatz der ausgefallenen Funktion durch eine andere Komponente, auf. Für die Fehlererkennung werden Duplication-And-Comparison, was dem klassischen 2-von-2-System entspricht, und Plausibilitäts-Checks, mit denen die Ergebnisse überprüft werden, eingesetzt.

Diese Ansätze werden u.a. bei COFTA [9] verwendet.

Hybride Redundanz kombiniert statische und dynamische Redundanz, um die Vorteile beider Ansätze, die niedrigeren Kosten der dynamischen Redundanz und die nicht vorhandene Recovery-Zeit der statischen Redundanz, nutzen zu können. Die verbreitetste Architektur hierzu sind m-von-n-Systeme mit zusätzlichen Komponenten, die ausgefallene Komponenten dynamisch ersetzen [25].

1. QS 9000: Quality System Requirements der US-Automobilindustrie - unterstützt durch ISO 9000.

Zur *Software-Redundanz* gibt es im wesentlichen zwei Ansätze, das N-Version-Programming und das Recovery-Block-Programming [26]. Beim N-Version-Programming [1] handelt es sich um einen statischen Ansatz. Mehrere diversitäre Programme, die aus der gleichen Spezifikation bzw. Problembeschreibung erstellt wurden, werden parallel ausgeführt und die Ergebnisse wie bei der statischen Hardware-Redundanz durch einen Voter bewertet. Beim Recovery Block Programming [33] werden ebenfalls verschiedene Versionen von Programmblöcken eingesetzt. Diese werden aber nicht parallel sondern sequentiell ausgeführt, solange bis eine der Ausführungen ein als korrekt bewertetes Ergebnis liefert. Die Fehlererkennung ist dabei z.B. durch Plausibilitätstests möglich. Eine Abwandlung dieses Mechanismus für einfache Controller stellen die RP-Actions (Actions for resilient processes) [34] dar.

Neben der vollständigen Fehlertoleranz gibt es noch weitere Ansätze, um Fehler zu behandeln. *Fail-safe*-Systeme stellen hier die einfachste Möglichkeit dar. Das System wird bei der Fehlererkennung aus dem Normalbetrieb in einen sicheren Systemzustand versetzt und angehalten. Bei *Fail-soft*- oder *Fail-operational*-Systemen wird ein Teil der Funktionalität auch nach einem Fehler aufrechterhalten. Dieses ist z.B. für einen Notlaufbetrieb notwendig, wenn ein sicherer Haltezustand nicht existiert.

Das Fail-safe Prinzip lässt sich auch auf einzelne Komponenten anwenden: Fail-Silent-Komponenten, die sich selbst überprüfen und bei einer Fehlererkennung in einen für das Gesamtsystem unkritischen Zustand übergehen, werden z.B. im X-By-Wire-Projekt [21] verwendet.

Literatur

- [1] A. Avizienis, *The N-Version Approach to Fault-Tolerant Software*, IEEE Trans. of Software Engineering, Vol. SE-11, No. 12, December 1985)
- [2] F. Balarin, M. Chiodo, A. Jurecska, H. Hsieh, A. L. Lavagno, C. Passerone, A. Sangiovanni-Vincentelli, E. Sentovich, K. Suzuki, B. Tabbara, *Hardware-Software Co-Design of Embedded Systems: The Polis Approach*, Kluwer Academic Publishers, 1997
- [3] R. E. Barlow, F. Proschan, *Statistische Theorie der Zuverlässigkeit: wahrscheinlichkeitstheoretische Modelle*, Verlag Harri Deutsch, Thun, 1978
- [4] L.B. Becker, M. Gergeleit, E. Nett, C.E. Pereira, *An Integrated Environment for the Complete Development Cycle of an Object-Oriented Distributed Real-Time System*, 2nd IEEE Int. Symposium on Object-oriented Real-time distributed Computing (ISORC'99), Saint-Malo, 1999
- [5] F. Belli, S. Pflieger, M. Seifert, *Software-Fehlertoleranz und -Zuverlässigkeit*, Springer-Verlag, Berlin, 1984

- [6] BMBF-Forschungsprojekt „KorSys: Korrekte Software für sicherheitskritische Systeme“,
<http://www4.informatik.tu-muenchen.de/proj/korsys/all/index.html>
- [7] O. Bringmann, W. Rosenstiel, A. Muth, G. Färber, F. Slomka, R. Hofmann, *Mixed Abstraction Level Hardware Synthesis from SDL for Rapid Prototyping*, Proceedings of the 10th IEEE International Workshop on Rapid Systems Prototyping (RSP'99), Clearwater, 1999
- [8] J.B. Bowles, *The new SAE FMECA Standard*, Proceedings Annual Reliability and Maintainability Symposium, 1998
- [9] B. Dave, N. Jha, *COFTA: Hardware-Software Co-Synthesis of Heterogeneous Distributed Embedded Systems for Low Overhead Fault Tolerance*, IEEE Trans. on Computers, Vol. 49, NO 4, April 1999
- [10] DFG-Schwerpunktprogramm „Entwurf und Entwurfsmethodik eingebetteter Systeme“,
<http://www.fzi.de/sim/people/hergen/sppes/sppes.first.html>
- [11] DFG-Schwerpunktprogramm „Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen“,
<http://tfs.cs.tu-berlin.de/projekte/indspec/SPP/>
- [12] DFG-Schwerpunktprogramm „Rapid Prototyping für integrierte Steuerungssysteme mit harten Zeitbedingungen“,
<http://www.ida.ing.tu-bs.de/research/coops/dfg-spp-rp/home.g.shtml>
- [13] *DIN V 19250 und 19251 zu MSR-Schutzeinrichtungen*, Beuth-Verlag, Berlin, 1994
- [14] *DIN V VDE 0801 zu Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben*, Beuth-Verlag, Berlin, 1990
- [15] *DIN VDI/VDE 3542 zu Sicherheitstechnische Begriffe für Automatisierungssysteme*, Beuth-Verlag, Berlin, 1998
- [16] *DIN 40041 zu Begriffe der Zuverlässigkeit*, Beuth-Verlag, Berlin, 1990
- [17] *DIN VDE 1000 zu allgemeine Leitsätze für das sicherheitsgerechte Gestalten technischer Erzeugnisse*, Beuth-Verlag, Berlin, 1979
- [18] *DIN EN 292-1 und 292-2 zu Grundbegriffe, allgemeine Gestaltungsleitsätze zur Sicherheit von Maschinen*, Beuth-Verlag, Berlin, 1991 bzw. 1995
- [19] E. Dönges, G. Färber, *Architekturvorschlag zur Entwicklung verlässlicher eingebetteter Realzeitsysteme auf Basis von Standardkomponenten*, at - Automatisierungstechnik 47, Oldenbourg Verlag, Juli 1999

- [20] G. Egger, *ESPRESS: Ingenieurmäßige Entwicklung sicherheitsrelevanter eingebetteter Systeme*, Manuskript zum Vortrag auf dem BMBF-Statusseminar Softwaretechnologie, Berlin, 1996
- [21] EU Brite-Euram-Projekt „*Safety Related Fault Tolerant Systems in Vehicles (X-By-Wire)*”,
Project No. BE 95/1329, Contract No. BRPR-CT95-0032
<http://www.vmars.tuwien.ac.at/projects/xbywire>
- [22] W. A. Halang, R. Konakovsky, *Sicherheitsgerichtete Echtzeitsysteme*, Oldenbourg-Verlag, 1999
- [23] S. Hariri, A. Choudhary, B. Sarikaya, *Architectural Support for Designing Fault-Tolerant Open Distributed Systems*, Computer, Vol. 25, No.6, June 1992.
- [24] P. Hummitzsch, *Zuverlässigkeit von Systemen*, Verlag Vieweg, Braunschweig, 1965
- [25] B. W. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley, Reading, Mass., 1989
- [26] M. Kersken (ed.), *Software Fault Tolerance*, Springer Verlag, Heidelberg, 1992
- [27] H.-D. Kochs, *Zuverlässigkeit großer und komplexer Systeme*, Technischer Bericht Nr. SI-11, Gerhard Mercator Universität Duisburg, 1994
- [28] O. Laitenberger. *Perspective-based Reading: Technique, Validation and Research in Future*. Technical Report, ISERN-95-01, 1995
- [29] J.-C. Laprie (ed.), *Dependability: Basic Concepts and Terminology*, IFIP WG 10.4 Dependable Computing and Fault Tolerance, Springer-Verlag, Wien, 1992
- [30] L. Litz. *Grundlagen der sicherheitsgerichteten Automatisierungstechnik*. at - Automatisierungstechnik 46, Oldenbourg Verlag, 1998
- [31] A. Österling, Th. Benner, R. Ernst, D. Herrmann, Th. Scholz, W. Ye, *Hardware/Software Co-Design: Principles and Practice*, Kapitel *The COSYMA System*, Kluwer Academic Publishers, 1997
- [32] S. Petters, A. Muth, T. Kolloch, T. Hopfner, F. Fischer, G. Färber, *The REAR Framework for Emulation and Analysis of Embedded Hard Real-Time Systems*, Proc. 10th IEEE Int. Workshop on Rapid System Prototyping RSP'99, 1999
- [33] B. Randell et al., *Reliability Issues in Computing Systems Design*, ACM Surveys, Vol. 10, No 2, June 1978

- [34] J. G. Silva, L. M. Silva, H. Madeira, J. Bernardino, *A Fault-Tolerant Mechanism for Simple Controller*, Proc. First European Dependable Computing Conference, Springer Verlag, Berlin, 1991
- [35] Sonderforschungsbereich 358 „Automatisierter Systementwurf“, <http://www.iee.et.tu-dresden.de/~sfb358/>
- [36] VDI, *Sicherheitstechnik und Automatisierung*, VDI-Bericht 1336, VDI-Verlag, Düsseldorf, 1997

3.4 Eigene Vorarbeiten

Erfahrungen und Vorarbeiten auf dem beantragten Gebiet wurden vor allem in Projekten des SFB 124 „VLSI-Entwurfsmethoden und Parallelität“ und im Laufe der ersten beiden Förderungsperioden dieses SFB erlangt bzw. durchgeführt.

Im Sonderforschungsbereich 124 hatten wir uns im Teilprojekt B4 “Chip Planning” unter anderem mit ECAD¹-Entwicklungsumgebungen beschäftigt. Obwohl wir uns dabei vor allem mit dem Entwurf hochintegrierter Schaltungen beschäftigten, findet man doch viele Parallelen zum Entwurf eingebetteter Systeme, dem Umfeld dieses Förderantrags. Unserer Meinung nach lassen sich die damals gewonnenen grundlegenden Überlegungen im Bereich der Entwurfstheorie [37], [43] - auf denen letztendlich die ECAD-Entwurfsumgebung PLAYOUT [45] basiert - auch auf den Entwurf eingebetteter Systeme verallgemeinern bzw. erweitern.

PLAYOUT [45] ist eine Entwurfsumgebung, bei der eine Vielzahl unterschiedlicher externer und selbstgeschriebener Entwurfswerkzeuge über eine zentrale Datenverwaltungs- und Entwurfsmanagement-Komponente gekoppelt sind. Ähnlich zum projektspezifischen Teil der im aktuellen SFB entwickelten Erfahrungsdatenbank werden in PLAYOUT alle Entwurfsdaten zentral verwaltet, wobei die Schnittstelle zur Datenverwaltungskomponente allen Entwurfswerkzeugen ihre eigene Sicht auf den Datenbestand ermöglicht.

Ein für diesen Antrag sehr wichtiger Aspekt bildeten die Arbeiten in Bezug auf ein interaktives Entwurfsmanagement. Basierend auf einer Analyse des aktuellen Entwurfszustandes und einer Abschätzung des weiteren Entwurfsverlaufs wird eine Planung der folgenden Entwurfsschritte ausgeführt und ein oder mehrere Entwerfer mit den nächsten Entwurfsaufgaben beauftragt. Die letztendlichen Entwurfsentscheidungen fällt allerdings der Mensch, so dass das PLAYOUT-Entwurfsmanagement eine Entscheidungsunterstützung darstellt und keinen vollautomatischen Entwurf anstrebt.

Die dem Entwurf zugrunde liegende Entwurfstheorie basiert auf dem Prinzip der schrittweisen Verfeinerung in einer zweidimensionalen Entwurfshierarchie [44]. Sie ist Grundlage der PLAYOUT-Produkt- und -Prozessmodellierung und wird zur Zeit

1. ECAD = Electronic CAD

im aktuellen SFB im Teilprojekt D1 und in projektübergreifenden Entwicklungsteams (Team 3.1 und 3.2) auch im Bereich des Entwurfs von Steuerungssystemen exemplarisch untersucht. Langfristiges Ziel dieses Teilprojektes ist es u.a., gemeinsam mit anderen Teilprojekten, ein Produkt- und Prozessmodell für alle Phasen des Entwurfs zuverlässiger eingebetteter Systeme basierend auf dem Prinzip der schrittweisen Verfeinerung zu entwickeln.

Praxisnahe Arbeiten im Bereich robuster, zuverlässiger Systeme wurden bei der Entwicklung zweier anwendungsspezifischer Parallelrechner durchgeführt. Gefördert im SFB 124 wurde im Teilprojekt B4 sowohl die Schaltung eines Parallelrechners zur effizienten Ausführung von Produktionssystemen [42] als auch ein weitgehend lauffähiger Prototyp eines Parallelrechners zur schnellen Ausführung von Verfahren aus dem Bereich der Bildverarbeitung (Echtzeit-Volumenvisualisierung) [39] entwickelt. Bei der ersten Rechnerarchitektur wurden in den frühen Phasen des Entwurfs Systemkomponenten, die in Bezug auf die Zuverlässigkeit des Systems kritisch werden könnten, identifiziert. Durch Einführung von Redundanz konnte anschließend die Fehlertoleranz erhöht werden. Bei der Entwicklung des Parallelrechners für die Bildverarbeitung wurden ebenfalls Untersuchungen durchgeführt, inwieweit die Verfügbarkeit der Architektur durch Redundanzbildung erhöht werden kann, die Ergebnisse flossen allerdings aus Aufwandsgründen nicht in den gebauten Prototyp ein.

Die Erfahrungen aus dem ECAD-Bereich sollen in dieses Teilprojekt einfließen. Dies betrifft neben dem allgemeinen Entwurfsansatz der schrittweisen Verfeinerung vor allem den Aspekt der Systempartitionierung und das Mapping von Systemfunktionalitäten auf Komponenten des verteilten Zielsystems. Hier möchten wir versuchen, die (Re-) Partitionierung von VLSI-Systemkomponenten [45] auf die Partitionierungsphase des Hardware-/Software Codesign auszudehnen. Ein Großteil der zur Partitionierung notwendigen Verfahren und Kostenfunktionen sind in den beiden Bereichen sehr ähnlich [40]. Die (nicht-funktionalen) Zuverlässigkeitsaspekte der zu entwerfenden Systeme betreffen in beiden Anwendungsbereichen einerseits den gesamten Entwicklungszyklus, werden andererseits schwerpunktmäßig in der Design-Phase in funktionale Anforderungen und Systemeigenschaften überführt.

Im aktuellen SFB wurden in den ersten beiden Förderungsperioden wesentliche Vorarbeiten im Teilprojekt D1, das vom Antragssteller mitgeleitet wurde, und über die Mitarbeit von D1 im Querschnittsprojekt Q1 geleistet. Hier wurde die Zuverlässigkeit exemplarisch vor allem in der Analysephase untersucht.

Während der ersten beiden Förderungsperioden wurden bei den projektübergreifenden Beispielenentwürfen die Aspekte Sicherheit und Zuverlässigkeit explizit in der Anforderungsanalysephase mit modelliert und berücksichtigt [38], [41]. Diese expliziten Betrachtungen der nicht-funktionalen Zuverlässigkeitsanforderungen waren wichtige Vorarbeiten in Bezug auf die schrittweise Überführung der Anforderungen in entsprechende Entwurfsentscheidungen in einem möglichst späten Entwicklungsschritt. Aus den Erfahrungen der Teamentwürfe lassen sich Hinweise für die Bestim-

mung geeigneter Entwicklungsphasen für einzelne Zuverlässigkeitsbetrachtungen ableiten.

Literatur

- [37] J. Altmeyer, B. Schürmann, *Design Formalization and Retrieval of Reuse Candidates*, J.S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design '96*, 231-250, Kluwer Academic Publishers, 1996
- [38] R. L. Feldmann, J. Münch, S. Queins, S. Vorwieger, G. Zimmermann, *Baselining a Domain-Specific Software Development Process*, SFB 501-Bericht Nr. 02/99, Universität Kaiserslautern, 1999
- [39] J. Lichtermann, *Eine Architektur zur Echtzeitvisualisierung von Volumendaten*, Dissertation, Fachbereich Informatik, Universität Kaiserslautern, 1998
- [40] S. Queins, T. Tetteroo, B. Schürmann, *Bewertung des dynamischen Verhaltens von SDL-Modellen*, SFB 501-Bericht Nr. 09/99, Universität Kaiserslautern, 1999
- [41] S. Queins, G. Zimmermann, *A First Iteration of a Reuse-Driven, Domain-Specific System Requirements Analysis Process*, SFB 501-Bericht Nr. 13/99, Universität Kaiserslautern, 1999
- [42] F. Schreiner, *PESA-I - A parallel distributed-memory architecture for production systems*, Dissertation, Fachbereich Informatik, Universität Kaiserslautern, 1992
- [43] B. Schürmann, *Modeling Design Data and Design Processes in the PLAYOUT CAD Framework*, Current Issues in Electronic Modeling (CIEM), Band 6, Kluwer Academic Publishers, Juni 1996
- [44] B. Schürmann, J. Altmeyer, M. Schütze, *On Modeling Top-Down VLSI Design*, Proc. IEEE Int. Conference on Computer Aided Design (ICCAD), San Jose, California, 1994
- [45] B. Schürmann, G. Zimmermann. *PLAYOUT - A Hierarchical Layout System*, in G. Zobrist (ed.) *VLSI Design Environments*, Gordon and Breach Science Publishers, S. 1-52, Amsterdam, 2000

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Eingebettete Systeme werden als korrekt erachtet, wenn sie ihrer Spezifikation genügen. Dies nachzuweisen ist eines der Anliegen vieler Teilaktivitäten im SFB. Hierbei wird die Annahme gemacht, dass sich die Umgebung der eingebetteten Software, d.h. die Hardware, die Benutzer und die Umwelt, wohl verhält. Wenn diese Annahme

jedoch wegfällt, womit man bei eingebetteten Systemen immer rechnen muss, so sollte das System nach Möglichkeit in der Lage sein, mit dem unerwarteten Fehlverhalten zurecht zu kommen. Man spricht hierbei von Robustheit und erhöhter Zuverlässigkeit.

Diese Thematik des Entwurfs robuster, zuverlässiger Systeme soll Thema dieses neu beantragten Teilprojekts sein. Hiermit werden zwei Aspekte, die bisher im SFB noch nicht ausreichend betrachtet wurden, angegangen. Zum einen soll anhand eines konkreten Effekts untersucht werden, wie nicht-funktionale Anforderungen systematisch in den Systementwicklungsprozess aufgenommen werden können. Dabei ist unser Ziel, durch die Trennung der funktionalen Anforderungen von den nicht-funktionalen (in diesem Fall speziell der Zuverlässigkeit) zu generischen, wieder-verwendbaren Prozessen und Modellen zu gelangen, die die zukünftige Entwicklung zuverlässiger eingebetteter Systeme wesentlich vereinfachen. Ein zweites umfassendes Ziel des Teilprojekts liegt in der Betrachtung der Zuverlässigkeit an sich. Sie ist unseres Erachtens nicht nur ein beliebiges Beispiel für nicht-funktionale Systemeigenschaften, sondern nimmt im Bereich der eingebetteten Systeme, wofür sich der SFB interessiert, neben den Zeitanforderungen eine herausragende Stellung ein. Bei der Entwicklung geeigneter Anwendungsentwicklungsprozesse sollte daher die Zuverlässigkeit nicht außer Acht gelassen werden.

Langfristiges Ziel des Teilprojekts ist die Betrachtung von Sicherheit und Zuverlässigkeit eingebetteter Systeme während des gesamten Entwurfprozesses. Sicherheit und Zuverlässigkeit haben Auswirkungen auf alle Phasen der Anwendungsentwicklung und bilden einen Schwerpunkt in der Design-Phase. In diesem Teilprojekt soll untersucht werden, wie bestehende Ansätze zur Entwicklung und Modellierung robuster, zuverlässiger und sicherer Systeme in den Systementwicklungsprozess einfließen können. Die Untersuchung formaler Aspekte der Design-Phase soll in enger Zusammenarbeit mit Teilprojekt C1 durchgeführt werden.

In den Mittelpunkt unserer Betrachtungen sollen die Bewertung der Verfügbarkeit und Funktionsfähigkeit entworfenen Systeme sowie die Erhöhung der Zuverlässigkeit durch Redundanz und Notlaufeigenschaften gestellt werden. Dabei sollen nicht (notwendigerweise) neue Bewertungsverfahren und Fehlertoleranzansätze entwickelt werden. Vielmehr wollen wir existierende Ansätze und Verfahren im Umfeld der Zuverlässigkeitsbetrachtung schrittweise in den Entwicklungsprozess des SFB einfließen lassen und so eine stufenweise Erhöhung der Zuverlässigkeit der im SFB entwickelten Systeme erreichen.

Aus den genannten Fragestellungen ergeben sich drei Aufgaben und Ziele, die im Folgenden beschrieben sind und deren Umsetzung in zwei Arbeitsschwerpunkten im Abschnitt 3.5.2 „Arbeitsprogramm“ erläutert wird. Unsere Ziele sind

- Verbesserung der Fehlertoleranz und Notlaufeigenschaften eingebetteter Systeme,
- Bewertung der Zuverlässigkeit,
- Untersuchung der Design-Phase.

Bei der Umsetzung dieser Ziele wird ein erster Arbeitsschwerpunkt die Erweiterung

des gesamten Systementwicklungsprozesses um die explizite Betrachtung von Zuverlässigkeitsanforderungen umfassen. Den zweiten Arbeitsschwerpunkt wird die eingehende Untersuchung der Design-Phase bilden. Hier haben die Sicherheits- und Zuverlässigkeitsanforderungen starken Einfluss auf die Systemstruktur, Komponentenauswahl und Redundanz.

Fehlertoleranz und Notlaufeigenschaften eingebetteter Systeme

Grundlage unserer Überlegungen sind die drei Ursachen für ein Systemversagen, d.h., dass die Komponenten eines eingebetteten Systems fehlerhaft entworfen sein können, dass Hardwarekomponenten (mit einer mehr oder weniger großen Wahrscheinlichkeit) ausfallen können und dass mit Bedienfehlern zu rechnen ist. Diesen Problemen muss bei der Forderung nach einem robusten und zuverlässigen System Rechnung getragen werden.

Der in der Regel nicht ganz zu erfüllenden Forderung nach einem ausfallfreien System wird durch zwei Verfahren begegnet. In Subsystemen, die aus Sicherheits- oder anderen Gründen nicht ausfallen dürfen, soll durch Einführung von Fehlererkennungsmaßnahmen und Redundanz die Fehlertoleranz erhöht werden. Um Kosten zu sparen, soll dies jedoch nur für ausgewählte, wichtige Teilsysteme erfolgen. Für das Gesamtsystem interessiert uns vielmehr die Frage, inwieweit die Robustheit und Fehlertoleranz des Systems durch die Einführung von Notlaufeigenschaften („stepwise degradation“, „fail-safe“ bzw. „fail-operational states“) erhöht werden können. Gedacht ist an eine Hierarchie von Betriebsfällen bezüglich des Verhaltens der verschiedenen Systemfunktionalitäten, beginnend mit einem „Normalbetriebsfall“, der das Idealverhalten des Systems charakterisiert, und endend mit einem „Minimalbetriebsfall“, der das absolute Minimum von Eigenschaften beschreibt (vgl. Abbildung 1). Für alle Betriebsfälle bzw. -modi müssen entsprechende Zuverlässigkeitsanforderungen spezifiziert und in geeignete funktionale Systemeigenschaften überführt werden. Ob Entwurfsentscheidungen alle Zuverlässigkeitsanforderungen erfüllen, muss durch geeignete Bewertungsverfahren nachgewiesen werden.

Bewertung der Zuverlässigkeit

Zur Bewertung der Zuverlässigkeit eines (teil-) entworfenen Systems müssen dessen Eigenschaften mit den Zuverlässigkeitsanforderungen verglichen werden. Hierbei sind einerseits stochastische Aussagen bzgl. der Ausfallwahrscheinlichkeit von Komponenten und Systemfunktionalitäten zu treffen und andererseits Auswirkungen von Fehlern auf das Systemverhalten zu untersuchen.

Die zuverlässigkeitsbedingten Systemanforderungen sollen für einzelne Aufgaben Angaben darüber enthalten, mit welcher Wahrscheinlichkeit eine Aufgabe erfüllt werden muss (→ Verfügbarkeitsanforderung) und eventuell darüber hinaus, durch welche (Notlauf-) Aufgabe sie im Fehlerfall ersetzt werden soll (→ Fail-Operational-Anforderung). Die erlaubte Ausfallwahrscheinlichkeit einer Aufgabe des Minimalbetriebsfalls liegt nahe bei Null und wird in Richtung Normalbetriebsfall ansteigen.

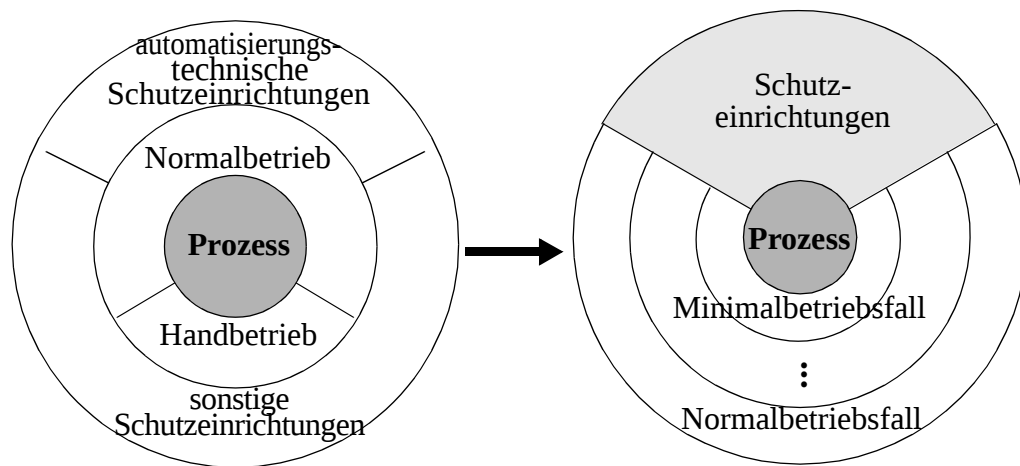


Abbildung 1: Einführung einer Hierarchie von Betriebsfällen
Links ist das in [30] beschriebene Schalenmodell zur Abgrenzung von Automatisierungsmaßnahmen dargestellt. Die beiden Betriebsfälle „Normalbetrieb“ und „Handbetrieb“ sollen durch eine Hierarchie von Betriebsfällen ersetzt werden.

Zur Auswertung der Verfügbarkeit werden physikalische Systemeigenschaften benötigt. Diese werden im Laufe der Systementwicklung in immer umfangreicherem Maße bekannt (der Großteil der Entscheidungen, die zu diesen physikalischen Eigenschaften führen, wird in der Design-Phase getroffen, weswegen diese Phase besonders betrachtet werden soll). Die Entwurfsqualität (→ Teilprojekt A1) und die bekannten physikalischen Systemeigenschaften sollen in ein geeignetes Fehlermodell einfließen, das den Entwurfsentscheidungen zugrunde gelegt bzw. mit dessen Hilfe die geforderte Verfügbarkeit überprüft werden kann. Das Fehlermodell wird entwurfsbegleitend verwendet, immer dann, wenn Entscheidungen über die physikalische Systemstruktur getroffen werden. Damit lassen sich Verletzungen bezüglich der Verfügbarkeit frühzeitig erkennen und Entscheidungen gegebenenfalls revidieren.

Zur Bewertung der Auswirkungen von Fehlern auf das Gesamtsystem sind Fehlerfortpflanzungsmodelle zu verwenden. Mit ihrer Hilfe hoffen wir, das Gesamtsystem so partitionieren zu können, dass sich Fehler möglichst nur lokal auswirken. Dies wird eine wesentliche Aufgabe der Design-Phase sein.

Ein wesentliches Ziel im Rahmen der Bewertung und Erhöhung der Zuverlässigkeit ist das Herausarbeiten generischer, wiederverwendbarer Aspekte. Langfristig sollen wiederverwendbare Modelle und Prozesse entwickelt werden, mit deren Hilfe sich die Zuverlässigkeit neu entwickelter eingebetteter Systeme verbessern lässt.

Design-Phase

In der zweiten Förderungsperiode wurde im SFB die Anforderungsanalysephase verstärkt untersucht. Mit den nun vorliegenden Ergebnissen soll in der dritten Förderungsperiode auch die Design-Phase intensiver betrachtet werden. Die Design-Phase ist für dieses Teilprojekt besonders wichtig, da in dieser Phase die nicht-funktionalen

Aspekte eine große Rolle spielen. Sie sind ausschlaggebend für die Systemstruktur (vor allem der hardwarenahen Schichten).

In der Design-Phase (Abbildung 2) ist das zu entwerfende System geeignet in Hardware- und Softwarekomponenten zu partitionieren. Diese Aufgabe fällt in das weite Feld des Hardware/Software-Codesign. Allerdings sind die Fragestellungen in unserem Fall von anderer Natur als bei den meisten anderen Untersuchungen. Es geht uns weniger darum, zeitkritische Aufgaben in Hardware zu bearbeiten, als vielmehr eine geeignete Laufzeitplattform für die Softwarekomponenten des Anwendungssystems zu finden. Die Zielarchitektur wird ein stark verteiltes System aus Controllern, Sensoren und Aktuatoren sein.

Die Fehlertoleranz ist in der Design-Phase durch Einfügen von Redundanz und Notlaufmaßnahmen den Anforderungen anzupassen. Hierzu sollen bekannte Verfahren zur Fehlererkennung, zum Einfügen von Redundanz bzw. Notlaufmaßnahmen und zur Bewertung der Zuverlässigkeit an die Architekturen der betrachteten Domäne

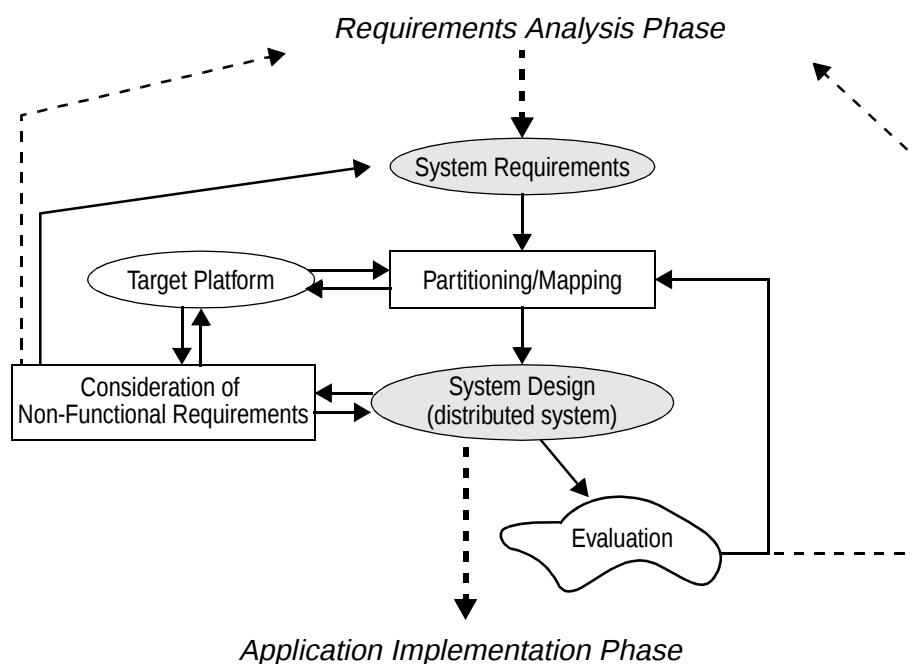


Abbildung 2: Vereinfachtes Prozessmodell der Design-Phase.

In einem iterativen Verfeinerungsprozess werden die (funktionalen) Systemanforderungen auf eine Zielplattform (Hardware und Systemsoftware) abgebildet. Nicht-funktionale Anforderungen werden schrittweise in strukturelle und funktionale Anforderungen/Eigenschaften umgesetzt, was die Systemanforderung und die Zielplattform verändert/erweitert. Die Bewertung eines Entwurfsschritts kann zu Iterationen in der Design-Phase, aber auch zu Rückschritten zu früheren Entwicklungsphasen führen.

Parallel zur Design- und Implementierungsphase der Anwendung erfolgt Design und Implementierung der Hardware sowie der Betriebs- und Kommunikationssysteme (Zielplattform).

angepasst werden. Langfristiges Ziel ist auch hier die Suche nach passenden generischen, wiederverwendbaren Modellen und Prozessen.

Insgesamt sollen folgende Fragen in diesem Teilprojekt beantwortet werden:

- Explizite Trennung funktionaler und nicht-funktionaler Anforderungen in den frühen Phasen des Systementwurfs und schrittweise Ersetzung der nicht-funktionalen Anforderungen in funktionale Anforderungen bzw. funktionale Systemeigenschaften.
- Geeignete Modellierung der Zuverlässigkeitsanforderungen und Notlaufeigenschaften in allen Systementwicklungsphasen. Festlegung, welche Entscheidungen im Hinblick auf die nicht-funktionalen Anforderungen wann getroffen werden müssen. Untersuchung der Wechselwirkungen zwischen dem Anwendungsentwurf einerseits und Systemstruktur, Hardware sowie Kommunikations-/Betriebssysteme andererseits. Daraus ergibt sich eine Erweiterung der Produkt- und Prozessmodelle sowie der Entwurfstheorie.
- Untersuchung der Überprüfbarkeit von Zuverlässigkeitsanforderungen mit Hilfe bekannter Fehlermodelle. Übernahme bekannter Maßnahmen zur Fehlererkennung und zum Einfügen von Redundanz in Hardware- und Softwarekomponenten zur Erfüllung der Zuverlässigkeitsanforderungen.
- Bearbeitung von Notlaufeigenschaften.
- Erarbeitung generischer, wiederverwendbarer Prozesse und Modelle.
- Sicherstellung der Verfolgbarkeit von Zuverlässigkeitsanforderungen.
- Einbettung des Ansatzes in den SFB-Entwicklungsprozess. Dies bietet die Möglichkeit der Werkzeugunterstützung - u.a. Test- und Validierungswerkzeuge. Darüber hinaus Einbindung der Zuverlässigkeitsbetrachtung in einen umfassenden Prototyping-Ansatz.

3.5.2 Arbeitsprogramm

Zur Umsetzung der oben genannten Ziele werden zwei Arbeitsschwerpunkte definiert. Im Arbeitsschwerpunkt A soll der bisherige SFB-Systementwicklungsprozess um den Aspekt der Zuverlässigkeit erweitert werden. Da die nicht-funktionalen Anforderungen im Wesentlichen die Design-Phase betreffen, wird diese im Arbeitsschwerpunkt B gesondert betrachtet. In beiden Arbeitsschwerpunkten sollen die Zuverlässigkeitsanforderungen zu robusten, fehlertoleranten Systemen führen, wobei die nicht-funktionalen Anforderungen im Laufe des Systementwicklungsprozesses schrittweise in funktionale Anforderungen bzw. Eigenschaften des Systems überführt werden.

Arbeitsschwerpunkt A: Erweiterter Systementwicklungsprozess

In diesem Arbeitsschwerpunkt soll der Entwurf zuverlässiger eingebetteter Systeme über alle Entwurfsphasen hinweg betrachtet werden. Hierzu werden zwei Aspekte vordergründig verfolgt, die in den beiden Arbeitspaketen A1 und A2 bearbeitet werden. Dies sind zum einen die Aufbereitung der Problembeschreibung zur Herausarbeitung der Zuverlässigkeitsaspekte, die in allen Entwurfsphasen schrittweise in funktionale Eigenschaften überführt werden können. Der zweite Aspekt ist die Einbindung der (um Zuverlässigkeitsaspekte erweiterten) Design-Phase in den gesamten Systementwicklungsprozess. Auch wenn der größte Teil der Zuverlässigkeitsaspekte die Design-Phase betreffen, soll in diesem Arbeitsschwerpunkt untersucht werden, welche Zuverlässigkeitsanforderungen in anderen Entwicklungsphasen bearbeitet werden sollten. Die Arbeiten haben starke Beziehungen zu der Entwicklung einer (Entwurfs-) Systematik, die im Teilprojekt D1 zusammen mit C1 ausgeführt werden soll. In einem dritten Arbeitspaket soll gemeinsam mit Teilprojekt D1 das Prototyping im Zusammenhang mit Zuverlässigkeit untersucht werden.

Arbeitspaket A1: Zuverlässigkeits- und Notlaufanforderungen

Schwerpunktmäßig sollen in diesem Arbeitsschwerpunkt die frühen Entwicklungsphasen betrachtet werden. Der Hauptaugenmerk lag im SFB bisher bei der Betrachtung der funktionalen Anforderungen während der Analysephase. Sicherheits- und Zuverlässigkeitsaspekte wurden exemplarisch mit den funktionalen Aspekten gleichbehandelt. In diesem Arbeitspaket sollen nun die funktionalen und nicht-funktionalen Aspekte voneinander getrennt werden, wobei im Bereich der Zuverlässigkeit eine hierarchische Modellierung von Notlaufeigenschaften („graceful degradation“) erarbeitet werden soll (vgl. Ziel „Fehlertoleranz und Notlaufeigenschaften“). Es soll untersucht werden, inwieweit der dem SFB zugrunde liegende objektorientierte Entwicklungsansatz es ermöglicht, in sich abgeschlossene Module zu konstruieren, deren Spezifikationen auch alle Fehlermöglichkeiten enthalten. Dazu soll die funktionale Spezifikation um notwendige Operationen für den Fehlerfall erweitert werden.

Personalbedarf: 8 PM

Arbeitspaket A2: Erweiterung der Entwurfsprozesse um Zuverlässigkeit

In diesem Arbeitspaket sollen die Ergebnisse des Arbeitsschwerpunkts B („Design-Phase“) auf den gesamten Entwicklungsprozess erweitert werden. Zum einen soll das Prozessmodell der Design-Phase in den Gesamtprozess eingebunden werden, zum anderen soll untersucht werden, welche Zuverlässigkeitsanforderungen in welchen Entwicklungsschritten genauer betrachtet werden sollten. Dieses Arbeitspaket kann erst in der zweiten Hälfte der Förderungsperiode, wenn erste Ergebnisse aus dem Arbeitsschwerpunkt B vorliegen, begonnen werden.

Ein langfristiges Ziel, das in diesem Arbeitspaket gelöst werden soll, ist in diesem Zusammenhang die Extraktion generischer, wiederverwendbarer Prozesse und Modelle aus durchgeführten Arbeiten anhand exemplarischer Entwürfe.

Personalbedarf: 6 PM

Arbeitspaket A3: Prototyping

Dieses Arbeitspaket ist eng verknüpft mit Arbeitspaket A2, da auch hier der gesamte Entwicklungsprozess im Vordergrund steht. Darüber hinaus liegt hier eine enge Kooperation mit Teilprojekt D1 vor, in dem unter anderem ein durchgängiger, entwerfsbegleitender Prototyping-Ansatz untersucht wird. Dieses Prototyping ist insbesondere solange wichtig, wie es noch nicht möglich ist, die gesamte Systementwicklung zu verifizieren.

Der Prototyping-Ansatz des SFB soll um das Phänomen Zuverlässigkeit erweitert werden. Dadurch erhält Prototyping eine neue Qualität. Durch die Systemsimulation anhand eines Prototypen können in Zusammenarbeit mit dem Kunden fehlende Notlaufforderungen festgelegt werden.

Personalbedarf: 4 PM

Arbeitsschwerpunkt B: Design-Phase

Der größte Teil aller Zuverlässigkeitsanforderungen wird während der Design-Phase in funktionale und strukturelle Systemeigenschaften umgesetzt. Daher bildet die Design-Phase einen eigenen Arbeitsschwerpunkt.

Zur Umsetzung der Zuverlässigkeitsanforderungen müssen diese mit den aktuellen Systemeigenschaften verglichen werden, wofür entsprechende Fehlermodelle ausgewertet werden müssen (Arbeitspaket B1). Ausgehend von der Systemanforderungsspezifikation und der Bewertung einer angenommenen Systemstruktur mit Hilfe der genannten Fehlermodelle muss eine modifizierte Spezifikation für die (restliche) Design-Phase hergeleitet werden (Arbeitspaket B2). Diese führt nach weiteren Entwurfsschritten (Partitionierung, Abbildung der Funktionalitäten auf Systemkomponenten) zu einem Entwurfsergebnis, das wieder bewertet werden muss. Die zuletzt genannten Schritte sollen in diesem Teilprojekt exemplarisch in Experimenten durchgeführt werden (Arbeitspaket B3).

Arbeitspaket B1: Bewertung der Zuverlässigkeit

Der Kern der Design-Phase bei der Entwicklung eingebetteter Systeme ist aufgrund der Vielzahl unterschiedlicher Optimierungskriterien ein iterativer Verbesserungsprozess. Nach jedem durchgeführten Design-Schritt muss das System bewertet und gegebenenfalls nach bestimmten Kriterien verbessert werden. Zur Bewertung der Zuverlässigkeit eines Entwurfs sollen existierende Modelle zur Anwendung kommen. Diese müssen ausgewählt und an die Produktmodelle des SFB angepasst werden. Hierbei müssen die Fehlermodelle, die in der Regel nur zwischen Betriebs- und Ausfallzuständen unterscheiden, auf die geplante Hierarchie von Notlaufeigenschaften angepasst werden.

Erarbeitete Ergebnisse dieses Arbeitspakets müssen anschließend auf den gesamten Entwicklungsprozess übertragen werden (vgl. Arbeitspaket A2).

Personalbedarf: 8 PM

Arbeitspaket B2: Umsetzung der Zuverlässigkeits- und Notlaufanforderungen
in strukturelle und funktionale Anforderungen

Die bis zum Beginn der Design-Phase nicht bearbeiteten Zuverlässigkeits- und Notlaufanforderungen werden in zwei Schritten in ein System-Design überführt. In einem ersten Schritt werden diese nicht-funktionalen Anforderungen in funktionale und strukturelle Anforderungen an das System transformiert. Diese Anforderungen betreffen einerseits die Fehlererkennung, andererseits Redundanz- und Notlaufaspekte. Das Ergebnis ist eine modifizierte Systemanforderungsspezifikation, auf der dann weitere Entwurfsentscheidungen aufsetzen. Einige der nicht-funktionalen Anforderungen sind an den Betriebssystem- und Kommunikationssystementwurf weiterzureichen bzw. in Kooperation mit diesen zu betrachten (→ Teilprojekte B4 und B5).

Personalbedarf: 6 PM

Arbeitspaket B3: Entwurfsumgebung und Design-Experimente

Nachdem alle nicht-funktionalen (Zuverlässigkeits- und Notlauf-) Anforderungen in funktionale und strukturelle Anforderungen überführt wurden (Arbeitspaket B2), ist eine geeignete Systemstruktur zu finden. Zentrale Operationen sind dabei die Partitionierung der (funktionalen) Objekte, die Festlegung einer geeigneten Hardware-Umgebung (soweit diese noch nicht festliegt) und die Abbildung der Funktionalitäten auf die Systemkomponenten. Diese Aufgaben fallen in das große Gebiet des Hardware/Software-Codesign und sollen in diesem Arbeitspaket exemplarisch in gezielten Experimenten durchgeführt werden. Das Ziel ist hierbei ein durchgängiger Prozess, der die Zuverlässigkeitsaspekte in geeigneter Weise berücksichtigt. Zur Durchführung der Experimente wird eine interaktive Entwurfsumgebung benötigt, die in diesem Arbeitspaket prototypisch entwickelt werden soll.

Gegen Ende der Förderungsperiode soll der erreichte Stand der Forschung durch ein größeres, teilprojektübergreifendes Experiment nachgewiesen werden.

Personalbedarf: 40 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausrüstung Torsten Tettersoo						
					AP A2	
	AP B1					
	AP B2					
	AP B3 - Implementierung					
	AP B3 - Experimente					
Ergänzungsausrüstung N.N.						
	AP A1					
					AP A2	
			AP A3			
	AP B3 - Implementierung					
		AP B3 - Experimente				
N.N., stud. Hilfskraft	AP B3					

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Das Teilprojekt bildet mit D1 eine Gruppe von Vorhaben, die sich die Aufgabe gestellt hat, verbesserte Systementwicklungsprozesse aus der gewählten Anwendungsdomäne zu entwickeln. Während D1 einen Entwicklungsansatz basierend auf Prototyping mit Schwerpunkt auf der Analysephase verfolgt, sollen in diesem Teilprojekt geeignete Verfahren zur Steigerung der Zuverlässigkeit der Anwendungssysteme in die Produkt- und Prozessmodelle des SFB integriert werden. Der Schwerpunkt von B11 liegt in der Design-Phase. Fragestellungen formaler Art im Rahmen der Design-Phase sollen gemeinsam mit C1 bearbeitet werden. Für Experimente soll die Testumgebung von D1 intensiv genutzt werden.

Bei der Untersuchung der Design-Phase im Allgemeinen und der Erweiterung um die nicht-funktionalen Zuverlässigkeitsanforderungen im Speziellen ist eine enge Zusammenarbeit mit B4 und B5 in Bezug auf eine geeignete Laufzeitplattform notwendig. Es ist zu untersuchen, welche Zuverlässigkeitsanforderungen durch die Anwendungssoftware zu berücksichtigen sind und welche in geeigneter Weise in Anforderungen an die Laufzeitplattform überführt werden.

Die in diesem Teilprojekt gewonnenen Verfahren und Modelle sollen langfristig in die zentrale SFB-Erfahrungsdatenbank einfließen. Dies erfordert eine Kooperation mit den Teilprojekten A1 und A3.

3.7 Ergänzungsausstattung für das Teilprojekt

	2001			2002			2003		
	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	1	104.400	BAT IIa	1	104.400	BAT IIa	1	104.400
	StdHkr.	1	20.400	StdHkr.	1	20.400	StdHkr.	1	20.400
	zus.:	2	124.800	zus.:	2	124.800	zus.:	2	124.800
SV	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
	(515)		-	(515)		-	(515)		-
	(522)		3.000	(522)		3.000	(522)		3.000
	zusammen		3.000	zusammen		3.000	zusammen		3.000
I	Mittel für Investitionen insgesamt			Mittel für Investitionen insgesamt			Mittel für Investitionen insgesamt		
	25.000			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)
SV: Sächliche Verausgabungsausgaben (Begründung vgl. 3.7.2)
I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Schürmann, B.; Dr. habil.	Techn. Inf.	FB Inf., Uni KL	5	1.1.95	
	(2) Tettersoo, T., Dipl-Inform.	Techn. Inf.	FB Inf., Uni KL	20	1.1.95	
Ergänzungsausrüstung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) N.N., (2) N.N., StdHkr.		FB Inf., Uni KL FB Inf., Uni KL	38,5 19	1.1.96 1.1.00	Bat IIa St. Hkr.
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Der Mitarbeiter der Grundausrüstung beschäftigt sich vor allem mit konzeptionellen Arbeiten im Bereich der Design-Phase und der Einbindung der Ergebnisse in den umfassenden SFB-Entwicklungsprozess. Bei der Entwicklung einer prototypischen HW/SW-Codesign-Umgebung zur Bearbeitung von Entwürfen und bei der Durchführung von Experimenten wird der Mitarbeiter vom wissenschaftlichen Mitarbeiter und von der studentischen Hilfskraft aus der Ergänzungsausrüstung unterstützt.

Von den zur Verfügung stehenden 18 PM fließen etwa zwei Drittel in die konzeptionellen Arbeiten, während ca. 6 PM für Implementierung und Experimente verfügbar sind.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Die konzeptionellen Arbeiten des Mitarbeiters der Ergänzungsausrüstung umfassen zunächst Erweiterungen des bisherigen SFB-Entwicklungsprozesses um Notlaufeigenschaften. Dies erfolgt in Zusammenarbeit mit den anderen Teilprojekten des SFB, die sich mit der Anwendungsentwicklung beschäftigen. Nachdem Ergebnisse aus dem Bereich der Design-Phase vorliegen, soll der Mitarbeiter gemeinsam mit dem Mitarbeiter der Grundausrüstung diese in den umfassenden Entwicklungsprozess integrieren. Für diese Arbeiten werden 18 PM veranschlagt.

Wesentliche Entwicklungstätigkeiten des Teilprojekts und ein Großteil der Experimente sollen vom Mitarbeiter der Ergänzungsausrüstung ausgeführt werden (18 PM), wobei er von der studentischen Hilfskraft der Ergänzungsausrüstung und vom Mitarbeiter der Grundausrüstung unterstützt wird.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausrüstung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausrüstung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausrüstung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausrüstung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
1 Workstation HP VISUALIZE C3600 mit 512 MB SDRAM; 1 X-Terminal	25.000	-	-
Summe	25.000	-	-

Die beantragte Workstation mit dem angeschlossenen Arbeitsplatz dienen zur Versorgung des im Rahmen der Ergänzungsausstattung beantragten Mitarbeiters und der studentischen Hilfskraft.

3.1 Allgemeine Angaben zum Teilprojekt B12

3.1.1 Thema

Stratified Component Frameworks

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Component Engineering

3.1.3 Leiter

Prof. Dr. Colin Atkinson

geb. 14.1.1962

Universität Kaiserslautern

Postfach 3049

67653 Kaiserslautern

Tel.: 0631/205-3958

atkinson@informatik.uni-kl.de

Dr. Thomas Kühne

geb. 9.11.65

Universität Kaiserslautern

Postfach 3049

67653 Kaiserslautern

Tel.: 0631/205-4654

kühne@informatik.uni-kl.de

Die Stelle des Leiters ist befristet: () nein (x) ja, bis zum 30.9.2001

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	() ja	(x) nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	() ja	(x) nein
Tierversuche	() ja	(x) nein
gentechnologische Untersuchungen	() ja	(x) nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Das Teilprojekt wird neu beantragt.

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
2001	145.200	5.000		150.200
2002	145.200	3.000		148.200
2003	145.200	3.000		148.200

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Components, frameworks, patterns and architectures have emerged as key technologies for promoting reuse in the development of software systems, and each is the subject of a great deal of ongoing research, especially within existing subprojects of the SFB 501. At present, however, it is not clear how they should be used together to achieve a maximal synergy of their respective strengths. The goal of this subproject is to investigate a practical approach for the integration of these technologies to promote reuse in the engineering of large, complex software systems. In doing so it will build upon ideas developed in previous SFB 501 subprojects, particularly those of B4, B5 and B10.

The key observation for achieving this synergy is that no single architecture (at no single level of abstraction) provides the optimal description of a system. On the contrary, different architectural presentations are optimal for different purposes. In traditional development projects, high-level (abstract) architectural descriptions produced during the refinement process are often discarded once the “real” architecture has been established. However, in framework-oriented software engineering approaches, where reuse is the prime concern, all architectural descriptions are valuable and must be carefully stored and related. We refer to a framework that contains a set of such architectural visualizations as a **stratified framework**. By using a component-oriented paradigm to construct the architectural strata in such a framework, and various kinds of patterns to establish the overall structure and relationships of the strata, an effective synergy of the aforementioned technologies can be attained. Furthermore, the tailorability and flexibility (and hence reusability) of the resulting frameworks is significantly improved. The key advantage of stratified frameworks over traditional frameworks is thus their support of reuse in terms of higher-level architectural representations of software systems.

The proposed project has four main subgoals -

- elaboration and description of the principles of framework stratification, and the properties of a stratified framework (the product),
- elaboration and description of a practical method for stratified framework development and application (the process),
- validation of the concept of framework stratification through the development of a case study (in the area of building automation),
- development of “stratification” tool support for the development and application of stratified frameworks.

In the first “research” phase, the technical and methodological foundations of the approach will be established. In the second phase, the basic premise of stratification will be validated and its key principles will be exercised. In the third phase a prototype supporting environment will be developed.

3.3 Stand der Forschung

Two time-honoured engineering strategies for tackling increased demand and complexity are **reuse** and **abstraction**, and over the years many techniques have been developed to apply these concepts in the context of software development. Today's leading-edge reuse technologies include -

- **components** - ready-to-run, prepackaged software capsules [1] [2]
- **frameworks** - preconfigured, semi-complete applications [3]
- **patterns** - proven solutions to common problems [4] [5]

while an abstraction technique currently attracting a great deal of attention is the concept of -

- **architecture** - a description of the fundamental structure and nature of a software system that ignores unnecessary details [6] [7]

Reuse and abstraction, in turn, are incarnations of an even more fundamental engineering principle - the **separation of concerns**. There is growing recognition in the research community that the key to improved software development practices is the strict and clean application of this principle. "Subject-Oriented Programming" [8], "Aspect-oriented development" [9], "Design Spaces" [10] and "Reflective (or meta-level) architectures" are all leading-edge development approaches motivated by the goal of highlighting and separate dimensions of concerns. Moreover, several workshops have recently sprung up with an explicit focus on separating concerns[11]. However, to date there has been little investigation of how the enhanced information and insights offered by these approaches should best be captured within frameworks to promote software reuse. The work proposed in this project aims to address this problem by investigating how to exploit the principle of separation of concerns in the development and application of component-based frameworks. In particular, it will investigate the impact of this principle on the architectural description of such frameworks.

The dominant "dimension of concern" under investigation in the project will be the "refinement" dimension, particularly the refinement of interactions. The importance of the explicit documentation of this dimension has been recognized in both formal and practical object-oriented development work. For example, Broy et al. [12] have developed a formal approach for modelling component-based systems in which the refinement of interactions plays a critical role in system structuring. Sounderajan [13], on the other hand, discusses the importance of interaction refinement in regular object-oriented software development. Stratification offers a way of systematically organizing and documenting the interaction refinements that underpin the different architectural perspectives of a system and often implicitly drive the development process.

The stratification concept also has a bearing on another important problem that is of great interest in the research community: namely, the definition of a clean model of "connectors" in the component-oriented description of system architectures [6], [3].

Most component and architecture description languages include the concept of “connectors” to capture the interaction protocols through which components, or architectural units, can be connected together. The intention is that these connectors should be “first class” citizens of the description languages that are amenable to the same set of manipulations as components themselves. The importance of supporting the first-class status of connectors results from the following observations and requirements:

- Traditional composition techniques, such as inheritance, are not suitable for dynamically “glueing” components together in a plug and play fashion
- In addition to a toolkit of readily available components, one should also possess standardized ways of combining them with a toolkit of connectors.
- Connectors can be complex, i.e., compositions of more primitive connectors.

To date, however, no entirely satisfactory model for connectors has been found. By viewing connectors as components (or collections of components) at lower architectural strata, a cleaner model for their access and manipulation may be possible. This, in turn, is expected to lead to enhanced opportunities for parameterization (i.e. “hot spot” definition). In conventional object-oriented frameworks, the “hot spots” (points of high-variation) are typically embodied as objects or classes. In other words, frameworks are typically parameterized in terms of objects within **one** architectural perspective (or strata). The object interaction mechanisms, in contrast, remain relatively fixed. However, it is well known that the functions (i.e. interactions in an object-oriented context) within software systems vary much more rapidly and significantly than objects. Indeed, this is the fundamental premise upon which the success of the object paradigm is based: objects are more stable than functions. By enabling the objects within the refinement of a high-level interaction to be points of variation (i.e. treated as “hot spots”) a stratified framework can offer more flexible parameterization than traditional object-oriented framework, and thus ultimately more flexible reuse. This will allow application engineers to influence interaction semantics, as well as object semantics, by defining specific objects tailored to the precise needs of the customer.

Since patterns will play an important role in capturing the relationships between strata, as well as the structure of the strata themselves, stratification also offers an approach for relating patterns within a “pattern language” [14]. This is regarded as the crucial step for realizing the full potential of the pattern approach. However, although architectural and design patterns are a very effective mechanism for capturing, motivating, and explaining proven design configurations, they only denote single spots in a design space and are not connected to each other very well. A pattern language, in contrast, is a generative hierarchy of patterns that narrows the choice for the application of patterns in terms of the context (i.e. the set of previously applied pattern) in which it is being applied. This recursive process is continued until the leaves of the pattern language have been reached. The patterns, and the way they are hierarchically connected, therefore guarantee the resulting design has an inherent quality. Currently, however, attempts to weave existing patterns into pattern languages are somewhat immature.

In summary, the research proposed in this thesis will make three fundamental contributions to the research focus of the SFB 501 -

- enhanced architectural abstraction mechanisms for promoting reuse in the development of large scale software systems,
- enhanced parameterization mechanisms for increasing the genericity of software frameworks, and through this, an enhanced model of “connectors”,
- enhanced pattern “languages” (or “systems”).

The resulting approach will therefore essentially offer an improved way of storing and reusing architectural knowledge in the development of software systems.

Literatur

- [1] C. Szyperski, *Component Software - Beyond Object-Oriented Programming*, Addison-Wesley /ACM Press, 1998.
- [2] L. Baum, *Towards Generating Customized Run-time Platforms from Generic Components*, Proc. of the 11th Conf. on Advanced Systems Engineering (CAISE-99), 6th DC, 14/15 June 1999, Heidelberg, Germany.
- [3] D. D'Souza and A. C. Wills, *Catalysis: Objects, Frameworks, and Components in UML*, Addison-Wesley, 1998.
- [4] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison Wesley, 1994.
- [5] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal M., *Pattern-Oriented Software Architecture: A System of Patterns*, John Wiley and Sons Inc. 1996.
- [6] Shaw, M., and Garlan D., *Software Architecture: Perspective on an Emerging Discipline*, Prentice-Hall, 1996.
- [7] L. Baum, M. Becker, L. Geyer, G. Molter, *The Role of Architecture for Complex Systems Development*, Proc. of 11th. Int'l Conf. on Software Engineering and its Applications 1998, 8-10 Dec. 1998, Paris, France.
- [8] Harrison W. and Ossher O., *Subject-Oriented Programming (A Critique of Pure Objects)*. In *Proceedings of OOPSLA'93*, pages 411-428, ACM 1993.
- [9] Kiczales, G., J. Lampling, A. Mendhekar, C. Maeda, V. Lopez, J.-M. Loingtier and J. Irwin, *Aspect-oriented Programming*, In *proceedings of the European Conference on Object-Oriented Programming*, Springer-Verlag LNCS 124, June 1997.

- [10] Baum, L., Geyer, L., Molter, G., Rothkugel, S., Sturm, P.: Architecture-Centric Software Development Based On Extended Design Spaces, Proc. 2nd ARES Workshop (Esprit IV 20477), February '98, Las Palmas
- [11] Workshop on Multi-Dimensional Separation of Concerns in Software Engineering (ICSE 2000).
- [12] Broy, M., Towards a Mathematical Concept of a Component and Its Use, TUM Report I9746, 1997.
- [13] Interaction Refinement in the Design of OO Systems, Sounderajan N., "OOPSLA'97 Workshop on Object-Oriented Behavioral Semantics", 1997.
- [14] James O. Coplien and Douglas C. Schmidt, (eds) "Pattern Languages of Program Design", Addison-Wesley, 1995.
- [15] Weiss, D. M., Lai, C. T. R. Software Product-Line Engineering. A Family-Based Software Development Process. Addison-Wesley, 1999.
- [16] Bayer, J., Flege, O., Knauber, P., Laqua, R., Muthig, D., Schmid, K., Widen, T., and DeBaud, J.-M. PuLSE: A methodology to develop software product lines. In Proceedings of the Symposium on Software Reusability (SSR'99), May 1999.
- [17] Moreira A. and S. Demeyer (Eds.), Object-oriented Technology, ECOOP'99 Workshop Reader, Lisbon Portugal, June 1999.
- [18] Feldmann R. L., Münch J., Queins S., Vorwieger S. and Zimmermann G., Baselineing a Domain-Specific Software Development Process. Technical Report No. 02/97, Sonderforschungsbereich 501, Dept. of Computer Science, University of Kaiserslautern, Germany, 1999.
- [19] Ergebnisbericht des Querschnittsprojekts Q3: "Erarbeitung eines Domänenmodells für das Anwendungsfeld Gebäudeautomation", Arbeits- und Ergebnisbericht 1998-2000, Sonderforschungsbereich 501, Fachbereich Informatik, Universität Kaiserslautern, 2000

3.4 Eigene Vorarbeiten

The roots of the stratification concept at the heart of the proposed project originated within the NASA MISSION project [20] [21] to help manage the complexity of large, non-stop distributed software systems such as the Space Station software. The complexity of the envisaged system called for several major distributed subsystems, including a distributed communication systems (DCS), a distributed information systems (DIS) and distributed policy management system (DPS). Architectural stratification proved to be the most effective way for visualizing the role and inter-

relationships of these major subsystems. Prof. Atkinson was a co-principal investigator of the project concerned.

The concept of stratification has been further refined within the AG Component Engineering, and was a focus of a working group at the Fourth International workshop on Component-Based Programming [22]. The working group concluded that stratification provided a valuable and effective way of visualizing the architecture of component-based systems [17].

Prof. Atkinson, co-principle investigator of the project, is also a leader of the Kobra project at the Fraunhofer Institute for Experimental Software Engineering. This project has developed a method for component-based product line engineering using the UML [23] [24]. Explicit separation of concerns is also a key principle of the Kobra approach, and framework development and application steps are clearly divided into well-defined dimensions. In contrast with the proposed project, however, the primary focus of Kobra is the composition dimension. Kobra components are modelled by suites of UML diagrams at a level of abstraction akin to analysis/design in regular development methods, and the refinement/translation of models into code is strictly separated from (de)composition of components into subcomponents [25]. The expertise gained from Kobra will be applied to the methodological issues in the proposed project. Moreover, since it will address a different dimension of concern, it is expected that the proposed project will provide a powerful complement to the separations already achieved in the Kobra approach.

T. Kühne, the other principle investigator on the project, has contributed to establishing the limitations of inheritance in software composition [29], and formulated the “Function Object” pattern [27] which is likely to be applicable in several areas of the proposed project. Not only is the introduction of function objects a recognised way to help “white-box” frameworks mature into “black-box” frameworks, but since function objects can be of “higher-order” they can facilitate certain forms of partial parameterization. As such, function objects can serve as combinators, just as in functional programming, and may thus provide one cornerstone of a clean model of component connectors.

The function object pattern is a part of a functional pattern system [28] which T. Kühne used to describe how functional paradigm can be embedded into object-oriented development. This pattern system will not only help to identify useful architectural styles for higher-level strata [26] but will also help to assess how patterns may further be empowered by a supporting stratification approach.

Literatur

- [20] C. W. McKay and C. Atkinson, "Supporting the Evolution of Distributed, Non-Stop, Mission and Safety Critical Systems," *Informatica*, vol. 19, no.1. 1995.

- [21] C. Atkinson and C. W. McKay, "A Generic Architecture for Distributed, Non-Stop, Mission and Safety Critical Systems," Second IFAC Workshop on Safety and Reliability in Emerging Control Technologies, Daytona Beach, FL, November 1-3.
- [22] C. Atkinson, T. Kühne, C. Bunse, Dimensions of Component-based Development, Fourth International Workshop on Component-Oriented Programming (WCOP'99; in conjunction with ECOOP'99), Lisbon, Portugal, 14 June 1999.
- [23] C. Atkinson, J. Bayer, O. Laitenberger and J. Zettel, Component-Based Software Engineering: The Kobra Approach, 2000 International Workshop on Component-Based Software Engineering, Limerick, Ireland, June 5-6, 2000.
- [24] C. Atkinson, J. Bayer and D. Muthig, Component-Based Product Line Development: The Kobra Approach, The First Software Product Line Conference, August 28-31, 2000, Denver, Colorado
- [25] C. Bunse and C. Atkinson, Improving the Quality of Object-Oriented Software: Systematic Refinement and Translation of Models to Code, Proceedings of the International Conference on Software and System Engineering and their Application, Paris, December 1999.
- [26] T. Kühne, Internal Iteration Externalized, Proceedings of 13th ECOOP, Portugal, June 1999.
- [27] T. Kühne, The Function Object Pattern, C++ Report, Vol 9 #9, pages 32--42, October 1997.
- [28] T. Kühne, A Functional Pattern System for Object-Oriented Design, Verlag Dr. Kovac, ISBN: 3-86064-770-9, July 1999.
- [29] T. Kühne, Parameterization versus Inheritance, Proceedings of TOOLS15, Australia, November 1994.

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

The main goal of the proposed project is to investigate, elaborate and support the concept of stratification in the creation and reuse of component-based frameworks. In doing so it will draw upon the main ideas developed in previous SFB 501 subprojects, as well as other leading edge work. At its core, a stratified framework is a partially preconfigured system organized in terms of a hierarchy of related, but distinct, architectural descriptions. This is represented schematically in Figure 1.

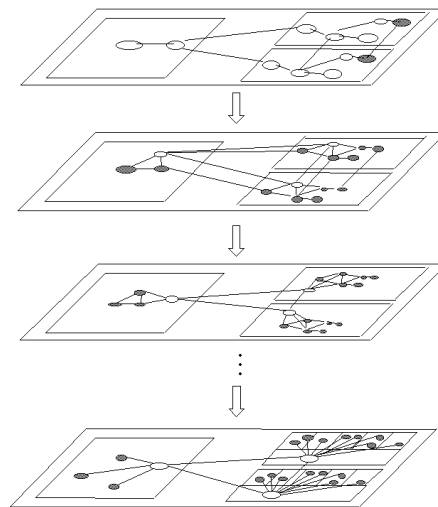


Abbildung 1: Architectural strata

In contrast with classic “layered” architectures, which essentially group the components within a system into related bands of abstraction (i.e. layers), the strata in a stratified architecture all represent **complete** descriptions of the system. The difference between the strata lies in their component population, and the level of abstraction at which the interactions between these components is described.

The relationship between the various strata is thus one of refinement. However, the focus of the refinement is the relationships (interactions) between components rather than components themselves. The refinement of an interaction at one level generally leads not only to multiple lower-level interactions, but also to the introduction of new components. This is depicted informally in Figure 2. The same situation also occurs in formal models of component-based systems. Figure 3 shows a similar situation but in the context of the formal component model of Broy et. al. [12].

In all object-oriented refinement models, the refinement of a high-level interaction generally leads to multiple lower-level interactions plus new lower-level components. With the stratification approach, however, instead of manipulating each refinement scenario independently, as is the case in typical development methods, individual refinements are reconciled and lower level components are consolidated to yield distinct levels of architectural representation. In effect, all individual refinements are handled within the context of the overall refinement lattice.

An important aspect of the proposed work is the distillation of a new form of pattern language (or system). By considering the entirety of refinement levels, one achieves a traceability between related component configurations residing at different levels which is usually lost when higher level strata are discarded. A stratified framework will thus allow the relationships between high level patterns (e.g., describing architectural styles) and other patterns at a lower level to be captured, but in a way that utilizes the typical component configuration mechanisms. Hence, investigating the

relationship of design patterns at different architectural strata will provide a way to “reverse-engineer” a pattern language from the typical, recurring application of design patterns. The rationale for this approach is the observation that proceeding down the patterns in a pattern language is akin to following the trace of a configuration of components at different architectural strata.

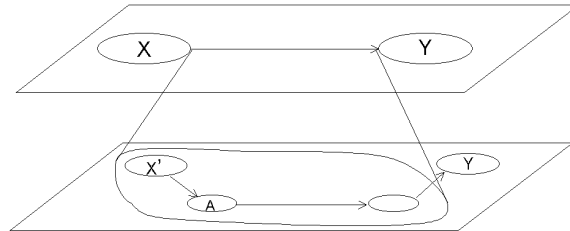


Abbildung 2: Informal model of interaction refinement

Another important research question to be addressed in the project is how to accommodate (de)composition of components within the framework. Conceptually (de)composition is an alternative dimension, but in practice it will have to be reconciled with the interaction refinement dimension and the resulting stratification levels.

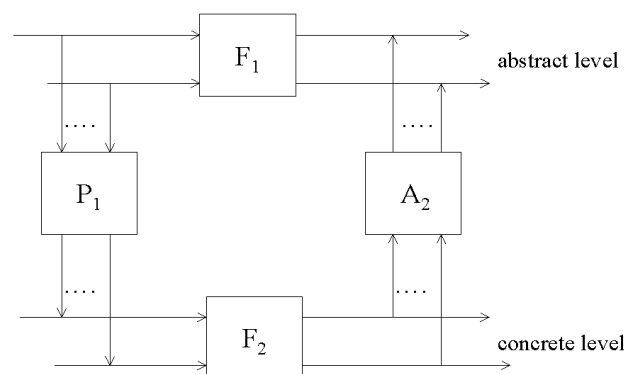


Abbildung 3: Interaction refinement: formal model [Broy]

The systematic and coherent visualization of interaction refinement is particularly problematic in object-oriented systems since the identity of the server object is bound up in the definition of the interaction. Part of the research work will therefore involve an investigation of the difference between function-oriented and object-oriented architectures from the perspective of stratification, and where appropriate the adaptation of leading function-oriented component approaches. In particular, the principle ideas from B4, B5 and B10 will be applied in an object-oriented context.

To accomplish these goals, the proposed project will have four primary areas of work:

- a) elaboration and description of the principles of framework stratification, and the properties of a stratified framework (the product),

- b) elaboration and description of a practical method for stratified framework development and application (the process),
- c) validation of the concept of framework stratification through the development of a case study (in the area of building automation),
- d) development of “stratification” tool support for the development and application of stratified frameworks.

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Methodological Foundations of Framework Stratification

The goal of this work area is to lay the methodological foundation for stratification and its practical use in the creation and application of frameworks. As with any engineering approach, two fundamental aspects of the method can be distinguished - the **product** and the **process**. The work area is thus divided into two work packages one focusing on the product and the other focusing on the process.

Arbeitspaket A1: Definition of the foundation principles (the product)

In this work package the concept of framework stratification will be consolidated and the key principles elaborated. The result will be a description of the product generated in a stratification project. The first activity in the work package will be the adaptation and integration of the relevant foundation technologies, especially the identified component models, architecture visualization techniques and/or patterns and interaction refinement model. Ideas from B5 and B10 are expected play an important role in this activity, as will the underlying conceptual models of leading component-based architectures and frameworks. The process will then be repeated to develop a strategy for capturing the relationships between strata. Ideas from B4 are expected to be of relevance here. Wherever possible, the approach will adopt practical, leading edge development technologies such as the UML and other OMG based-standard, such as the Object Management Architecture (OMA). Among other things, therefore, this work package will provide an object-oriented application of the key ideas from previous SFB 501 projects using important emerging object-oriented standard such as the UML and CORBA.

Personalbedarf: 12 PM

Arbeitspaket A2: Definition of the process for developing and applying Stratified Frameworks.

Once the characteristics of a stratified framework have been defined, the process for creating stratified frameworks and for using them in the creation of software systems can be defined. This is the goal of work package A2. As in the previous work package, the first activity will involve the adaptation and integration of relevant foundation technologies. The key challenge lies not so much in the definition of techniques

for elaborating the structure of an individual strata, since this is expected to rely heavily on existing patterns and architectural styles, but rather in identifying **what** strata are the most appropriate for a given domain, and how they should be inter-related. So far as possible, the process will be based on the existing SFB 501 process for software development [18]. In addition, “separate of concern” techniques such as aspect-oriented development [9], reflective architectures and extended design spaces (B10) [10] will be investigated. Techniques for framework scoping and generalization will also be adapted from well known product-line approaches such as PuLSE [16] and FAST [15]. The investigation of separation of concerns will also consider the role of component (de)composition and its relationship to architectural strata. The contribution of stratification to pattern language (system) formulation will be investigated in this work package.

Personalbedarf: 18 PM

Arbeitsschwerpunkt B: Case Study

The basic goal of this work area is to validate the approach developed in work area A by means of a case study. In accordance with the general direction of SFB 501, the case study will lie in the area of building automation. Framework-oriented development of software families is broadly speaking a two step process: the first step is the generation of a generic framework which is generalized to cover at least the common needs of systems in the domain, and the second step is the instantiation of the generic framework for a specific system. Work area B is divided into two subpackages which roughly corresponding to these two activities.

Arbeitspaket B1: Stratified Framework Development

The goal of this package is to develop a prototype stratified framework. Since frameworks, by their nature, are intended to provide extensive coverage of the domain of interest (building automation in this case) it will not be possible to create a full-scale framework with a complete complement of components, or even of architectural strata. The goal of this work package is rather to build a cut-down prototype version of a framework in order to demonstrate the feasibility of the approach, and to illustrate the nature of the components and strata that will result in the stratification approach. The aim therefore, will be to define a self-contained subset of a full-scale framework which can be carried through to implementation and ultimately running code.

The key activities that will be carried out during this activity will be the analysis and scoping of the domain, the identification and consolidation of architecture strata, the selection, analysis and modelling of self-contained subset of components, and the translation of the invariant parts of these components into a suitable implementation vehicle. The first of these activities, domain analysis and scoping, will draw heavily on the domain analysis work carried previously in the initial phase of the SFB 501 [19]. The goal will be to identify a system boundary which fits closely with previous SFB 501 case-studies, and thus facilitates optimal comparison of approaches

To maximize the dissemination and potential adoption of the approach, the case study will focus on the use of leading edge, object-oriented development and execution technologies. In particular, the UML will be used at the analysis/design level, Java will be used as the programming language and JavaBeans and/or CORBA will provide the underlying component technology. As well as demonstrating/validating the principles of stratification, therefore, this project will provide a demonstration of key ideas from B5 and B10 in the context of leading edge object-oriented/ component-based development technologies. Both will require care in the selection of representative strata, and the implementation of representative components.

Personalbedarf: 18 PM

Arbeitspaket B2: Framework Instantiation

The goal of this work package is to perform an illustrative instantiation of the framework to yield working code. A representative set of application features will be selected, and the corresponding components coded to provide the desired functionality. As with the development of the framework itself, the underlying goal will be to demonstrate as many of the advantages of stratification as possible, particularly the increased ease of instantiation and adaptation that is expected. This means that system behaviour and features will be selected so as to require the instantiation of components at various strata in the framework.

Personalbedarf: 12 PM

Arbeitsschwerpunkt C: Tool Development

The essence of the stratification idea is the explicit storage and manipulation of entities at all meaningful levels of abstraction (much of which is often discarded today), and the explicit storage and manipulation of the relationships between these entities. In traditional software engineering methods, this information is often referred to as traceability information. The stratification approach, therefore, has much to gain from the practical support of a dedicated tool. The purpose of this work package is to develop a prototype stratification tool. Three main work packages are envisaged for this task: Requirements Specification, Meta-Model Development and Software Development

Arbeitspaket C1: Requirements Specification

The goal of this work package is to define the requirements for a stratification tool, first for an ideal tool, and second for the specific prototype to be constructed during the project. The requirements specification will include definitions of the functional requirements, the non-functional requirements and the logical interface. A mixture of textual documents and UML diagrams will be used.

Personalbedarf: 12 PM

Arbeitspaket C2: Meta Model Development

Since the tool is essentially an information processing tool, a metamodel will be defined to capture the essence of the concepts to be stored and manipulated by the tool. This meta-model will not only serve as the basis for the implementation of the tool, but will also help provide a concise, abstract description of the key concepts of stratification. Established UML-oriented techniques for metamodel development and documentation will be utilized.

Personalbedarf: 10 PM

Arbeitspaket C3: Tool Construction

In the final work package a working prototype will be generated. A standard three-phase approach will be adopted, involving the development of an architecture, a detailed design and code. UML will be used for the description of the architecture and the detailed design, and Java will be used as the implementation language.

Personalbedarf: 8PM

3.5.3 Zeitplan

Personal	2001 1. H. J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausstattung						
Kühne (20 Std.)	A1		A2		B1	B2
Ergänzungsausstattung						
NN (38,5 Std.)	A1	A2		B1	B1/B2	C2
HW1 (19 Std.)	B1	B2	C1		C2/C3	C3
HW2 (19 Std.)	B1	B2	C1		C2/C3	C3

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

The central goal of the SFB 501 is the investigation of generic techniques for the cost-effective engineering of large software systems. This project focuses directly on this goal in the context of mainstream, leading-edge, object-oriented technologies; namely components, patterns, architectures and the UML. If successful, the developed approach will offer a way of combining the advantages of these individual technologies to better support reuse. Moreover, the larger the system, the greater the expected benefits.

The proposed project will build upon ideas developed in the first round subprojects in SFB 501, namely subprojects B4, B5 and B10. As well as integrating results from

these projects, therefore, the project will also apply some of their key ideas in the context of leading-edge object-oriented methods and component-based platforms. The proposed project will also build upon the standard SFB 501 software development process, and will reuse prior work on the analysis of the SFB 501 case study domain.

Given the increased use of distributed systems, communication support is likely to play an important role in most large software systems. From the perspective of stratified architectures, this means that at least one of the levels in a stratified framework is likely to be oriented towards the description of the communication subsystem. Communication architectures developed within the subproject B4 will serve as the basis for the communication level architectural descriptions which will be needed in the case study. Also, the “protocol” patterns developed in B4 are likely to be valuable in developing the interaction refinements at (or close to) the communication stratum.

The underlying paradigm under investigation in the project is the component paradigm. This means that each architectural stratum essentially represents a component-based architecture focusing on a certain aspect of a system’s realization. The organization and description of each architectural level will therefore draw on the architectural concepts and generic component concepts developed in subproject B5.

The key to the successful stratification of frameworks is the careful identification, architecting and positioning of the different strata in the framework. This essentially involves the crystallization of the optimal dimensions of concern for the system(s) in hand. Several development paradigms address this issue from different angles, but one of the potentially most effective is the design spaces concept elaborated within the subproject B10. In particular, the notion of extended design spaces will be investigated as a strategy for identifying and structuring architectural strata. Whereas B10 focuses on the use of design spaces for the construction of a single architecture, however, B12 will investigate the approach’s potential for supporting the definition of multiple architecture visualizations, and for explicitly defining the nature of the relationships between them.

3.7 Ergänzungsausstattung für das Teilprojekt

	2001			2002			2003		
	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	1	104.400	BAT IIa	1	104.400	BAT IIa	1	104.400
	StdHkr.	2	40.800	StdHkr.	2	40.800	StdHkr.	2	40.800
	zus.:	3	145.200	zus.:	3	145.200	zus.:		145.200
SV	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
	(522)		5.000	(522)		3.000	(522)		3.000
	zusammen		5.000	zusammen		3.000	zusammen		3.000
I	Mittel für Investitionen insgesamt			Mittel für Investitionen insgesamt			Mittel für Investitionen insgesamt		
	-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)
SV: Sächliche Veraulungsausgaben (Begründung vgl. 3.7.2)
I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Atkinson, C.; Prof. Dr. (2) Kühne, K., Dr.	Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL	5 20		C3 IIa
Ergänzungsausrüstung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) N.N., Dipl-Inform. (2) N.N., StdHkr. (3) N.N., StdHkr.	Prakt. Inf. Prakt. Inf. Prakt. Inf.	FB Inf., Uni KL FB Inf., Uni KL FB Inf., Uni KL	38,5 19 19		IIa
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

T. Kühne will work on the definition of the foundation principles of the stratified framework concept and will work on the development of a process to create and apply stratified frameworks. He will furthermore, contribute to and consolidate the results of developing a prototype stratified framework for a domain fitting closely with previous SFB 501 case-studies.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

This Mitarbeiter will contribute to the definition of the process for developing and applying Stratified Frameworks. and then focus on the validation of the results achieved in work area A by means of a case study. He will also contribute to the conceptual foundations for a tool design.

Both studentische Hilfskräfte will perform preparatory work for workpackage B and then focus their attention to the conception and implementation of a prototype stratification tool.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausrüstung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausrüstung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausrüstung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	5.000	3.000	3.000

Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausrüstung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
	-	-	-
Summe	-	-	-

Projektbereich C

Beschreibungstechniken

3 Projektbereich C: Beschreibungstechniken

3.01 Zusammenfassende Darstellung des Projektbereiches

In diesem Projektbereich werden Beschreibungstechniken untersucht, die zur maschinellen Erfassung aller relevanten Aspekte der im SFB für die Anwendungsdomäne zu entwickelnden Modelle geeignet sind. Die Beschreibungstechniken sollen insbesondere die zentralen Aspekte Verfolgbarkeit, Generizität und Wiederverwendbarkeit unterstützen. Die Teilprojekte befassen sich mit folgenden Themenbereichen:

- a) Klärung der Begriffswelt und Erarbeitung eines Basisreferenzmodells, das die zentralen Konzepte der Softwareentwicklung erfaßt.
- b) Bewertung, Anpassung, Erweiterung, Koordinierung und Integration der unterschiedlichen Beschreibungstechniken, die im SFB Verwendung finden.
- c) Entwicklung eines Rahmenmodells zur Integration formaler und nichtformaler Beschreibungen von Produkt-, Prozeß- und anderem Wissen zum Zwecke des Verstehens, Generierens, Verifizierens, Validierens und Modifizierens von Systemen im Kontext des Anwendungsbereichs.

Bei der Entwicklung großer Systeme ist die Verwendung von sehr unterschiedlichen Beschreibungstechniken erforderlich, nicht nur wegen der notwendigen Abstraktionsstufen, sondern auch wegen der unterschiedlichen logischen Sichten auf und Anforderungsarten an ein System. Die zur Zeit verwendeten Beschreibungstechniken reichen von natürlichsprachlichen über informelle und halbformale - oft graphische - Beschreibungstechniken bis hin zu einer Vielzahl von formalen Beschreibungstechniken mit wohldefinierter Syntax und Semantik. Die Zusammenhänge zwischen den einzelnen Beschreibungen sind oft vage und die Kluft zwischen ihnen meistens zu groß.

Produkte der Softwareentwicklung müssen somit als Geflechte von Dokumenten gesehen werden, die unterschiedlichen Zwecken dienen, und ein System und seine Eigenschaften beschreiben. Produktmodelle legen fest, welche Form ein Produkt hat, was in den einzelnen Dokumenten und im Produkt beschrieben wird und in welcher Form die Beschreibung stattzufinden hat. Sie bilden die Grundlage zur Erfassung von Softwareentwicklungsprozessen, die Produkte erzeugen. Die Festlegung der einzelnen Prozesse, ihre Beschreibung und Kompositionalität geschieht durch Prozeßmodelle. Eine Formalisierung und Integration dieser Modellwelten sowie die geeignete Instanziierung auf das Anwendungsgebiet ist von grundlegender Bedeutung für die Untersuchungen im SFB. Besondere, oft informell beschriebene Richtlinien und Regeln für die Entwicklung und Reihenfolge der Entwicklung einzelner Produkte legen Vorgehensweisen fest, die durch Vorgehensmodelle erfaßt werden. Sie bilden die Grundlage für die Planung von Projekten für die Entwicklung großer Systeme. Diese Modelle sollen im SFB entwickelt und möglichst formal gefaßt werden. Der C-Bereich ist an dieser Modellbildung maßgeblich beteiligt.

3.02 Bisherige und beantragte Förderung des Projektbereiches

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	645,2	6,0	-	651,2
1998	100,8	3,0	-	103,8
1999	103,2	3,0	-	106,2
2000	104,4	3,0	-	107,4
Zwischensumme	953,6	15,0	-	968,6
2001	249,6	3,0	-	252,6
2002	249,6	3,0	-	252,6
2003	249,6	3,0	-	252,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000,-)

3.1 Allgemeine Angaben zum Teilprojekt C1

3.1.1 Thema

Formale Beschreibungstechniken

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, theoretische Informatik

3.1.3 Leiter

Prof. Dr. Jürgen Avenhaus
geb. 19.08.1941
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2633
Fax: 0631/205-2156
avenhaus@informatik.uni-kl.de

Prof. Dr. Klaus Madlener
geb. 15.01.1947
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2268
Fax: 0631/205-2156
madlener@informatik.uni-kl.de

Die Stelle des Leiters ist befristet ☒ ja ☐ nein () ja, bis zum

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ ja	☒ nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ ja	☒ nein
Tierversuche	☐ ja	☒ nein
gentechnologische Untersuchungen	☐ ja	☒ nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	645,2	6,0	-	651,2
1998	100,8	3,0	-	103,8
1999	103,2	3,0	-	106,2
2000	104,4	3,0	-	107,4
Zwischensumme	953,6	15,0	-	968,6
2001	249,6	3,0	-	252,6
2002	249,6	3,0	-	252,6
2003	249,6	3,0	-	252,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

Bei der Entwicklung großer Systeme werden in zunehmendem Umfang formale Beschreibungstechniken eingesetzt. Damit werden sowohl einzelne (Teil-)Produkte als auch der Entwicklungsprozess beschrieben. Hierzu wird eine Vielzahl verschiedener Beschreibungstechniken verwendet, deren Zusammenhang jedoch noch nicht ausreichend geklärt ist.

Das Teilprojekt verfolgt auch in Zukunft das Ziel, formale Beschreibungstechniken im Kontext der Anwendungsdomäne zu untersuchen und die übrigen Teilprojekte bei der Verwendung derartiger Techniken zu unterstützen.

In der ersten Förderungsperiode stand die Untersuchung und gezielte Integration von Produktbeschreibungssprachen sowie die Einarbeitung in die Anwendungsdomäne durch Mitarbeit in den Teams im Vordergrund. In der zweiten Förderungsperiode wurde zum einen die Integration von Produktbeschreibungssprachen fortgesetzt. Zum anderen wurde aufbauend auf den in der ersten Förderungsperiode gemachten Erfahrungen insbesondere die methodische Verwendung existierender formaler Beschreibungstechniken im Bereich der Anforderungen untersucht und verbessert. Hierzu wurde ein Referenzmodell entwickelt, das unabhängig von einer bestimmten Beschreibungstechnik den Inhalt eines Dokuments festlegt, das wir *Problemspezifikation* nennen. Des Weiteren wurde dieses Referenzmodell für eine Temporallogik instanziiert, die mit maßgeschneiderten Operatoren angereichert und mit Strukturierungskonzepten wie Modularisierung, Aggregation, Vererbung und Parametrisierung erweitert wurde.

Aufbauend auf diesen Ergebnissen wird in der dritten Förderungsperiode einerseits die Eignung des bisher entwickelten Referenzmodells und seiner Instanzierung weiter überprüft und die methodische Unterstützung bei der Erstellung von Problemspe-

zifikationen verbessert. Andererseits steht der Übergang von Problemspezifikationen hin zu Entwurfsspezifikationen im Mittelpunkt der Untersuchungen.

Zentrale Fragestellungen hierbei sind: Wie kann die Erstellung sowohl von Problem- als auch von Entwurfsspezifikationen durch *Kompositionsoperatoren* geeignet unterstützt werden? Welche Arten von *Verfolgbarkeitsrelationen* werden horizontal, d.h. innerhalb einer Problem- oder Entwurfsspezifikation, sowie vertikal, d.h. zwischen einer Problem- und einer Entwurfsspezifikation benötigt und wie können diese angemessen dargestellt werden? Welche Informationen - u.a. *Verhalten* und *Architektur* - werden in einer Entwurfsspezifikation repräsentiert und welche Beschreibungstechniken sind hierfür geeignet?

3.3 Stand der Forschung

Ein zentrales Ziel, das das Teilprojekt C1 mit der Entwicklung eines Referenzmodells für die Software-basierte Entwicklung großer, reaktiver Systeme in den Arbeitsschwerpunkten A und B verfolgt, ist die mathematisch präzise Erfassung vieler Begriffe und ihrer Zusammenhänge im Bereich des Software-Engineering, die ansonsten sehr häufig unpräzise und mehrdeutig verwendet werden. Das Problem einer fehlenden mathematisch präzisen Fundierung, insbesondere im Zusammenhang mit UML [1], wird von einer großen Anzahl von Personen thematisiert, siehe zum Beispiel [2], [3], [4] und die nachfolgend angegebenen Arbeiten.

Während vor allem in [5], aber auch in [6] auf die Gefahren hingewiesen werden, die die Verwendung einer semantisch nicht präzise fundierten Beschreibungstechnik haben kann, werden in [7] und [8], unter anderem mit den Arbeiten der *precise UML Gruppe*, Ansätze vorgestellt, die darauf abzielen, für UML eine mathematisch präzise semantische Grundlage zu entwickeln.

Viele der heute existierenden Ansätze zur Formalisierung objekt-orientierter Konzepte, wie zum Beispiel der Aggregation oder der Vererbung, basieren auf der Kategorientheorie [9], [10], [11]. Dies scheint uns nicht geeignet, um die Formalisierung einem größeren Personenkreis zugänglich zu machen. Deswegen haben wir entsprechende Formalisierungen ausschließlich auf der Grundlage mengentheoretischer Konzepte durchgeführt und wollen dies so auch weiterhin tun.

Unser bisheriges Referenzmodell ist von den Arbeiten von Zave und Jackson geprägt, siehe [12], [5].

Eine mathematisch präzise Erfassung der im Software-Engineering verwendeten Begriffe ist auch eine entscheidende Voraussetzung für jede Art von Werkzeugunterstützung. Für die Erfassung und Dokumentation von Anforderungen verweisen wir exemplarisch auf die in den Gruppen von Parnas [13] und Heitmeyer [14] entwickelten Werkzeugkästen für tabellenbasierte Beschreibungstechniken.

Hinsichtlich der Architekturbeschreibungssprachen (ADL) verweisen wir wegen der inzwischen großen Anzahl existierender ADLs auf den Überblicksartikel [15].

Arbeitsschwerpunkt C beschäftigt sich mit der Verfeinerung von ASM-Entwurfsspezifikationen. In [16] werden Verfeinerungs- und Kompositionsrelationen zwischen Transitionssystemen angegeben. Hierbei liegt der Schwerpunkt der Untersuchungen auf der mathematischen Verifikation von Compilern. In [17] werden Kompositions- und Strukturierungsprinzipien für ASMs vorgestellt, etwa sequentielle Komposition, Parametrisierung und Fehlerbehandlung. In [18] wird am Beispiel des Bakery-Algorithmus (L. Lamport) Verfeinerung zwischen verteilten ASMs betrachtet. Hierbei ist die zu Grunde liegende Verteilung durch so genannte partiell geordnete Läufe (true concurrency) gegeben. In Arbeitsschwerpunkt C soll u.a. geklärt werden, was unter einer ASM-Entwurfsspezifikation verstanden werden soll. Hierzu ist eine eingehende Untersuchung von Kompositionstechniken notwendig. Innerhalb der ASM-Methode gibt es hierzu bisher nur wenige Ergebnisse (etwa [16] und [17]). Ein zentrales Anliegen von Arbeitsschwerpunkt C ist die Untersuchung von Verfeinerungsrelationen. Hierbei soll vor allem eine auf Kausalität basierende Verteilung (true concurrency) der Spezifikation zu Grunde gelegt werden. Dies ist wichtig, da durch eingeschränktere Verteilungsmodelle (etwa interleaving) zusätzliche Abhängigkeiten implizit spezifiziert werden, die nicht ursächlich dem jeweiligen Problem zuzuordnen sind. Innerhalb ASMs wird echte Verteilung durch so genannte partiell geordnete Läufe modelliert. Es gibt bisher nur wenige Arbeiten im ASM-Kontext (etwa [18]), die Verfeinerung und die darauf aufbauende Verifikation für diese Art der Verteilung thematisieren.

- [1] G. Booch, J. Rumbaugh, I. Jacobson: *The Unified Modeling Language User Guide*. Addison Wesley, 1999.
- [2] J. J. Odell, G. Ramackers: *Toward a Formalization of OO Analysis*. In J. J. Odell, *Advanced Object-Oriented Analysis & Design Using UML*, S. 43-55, Cambridge University Press, 1998.
- [3] M. Broy, O. Slotosch: *Enriching the Software Development Process by Formal Methods*. In D. Hutter, W. Stephan, P. Traverso, M. Ullmann, *Applied Formal Methods - FM-Trends 98*, Springer, 1998.
- [4] M. Heisel: *Softwaretechnik und Methodik*. In *Antrittsvorlesungen der Fakultät für Informatik*, Universitätsschriften der Universität Magdeburg, 1998.
- [5] M. Jackson: *A Discipline of Description (Keynote Talk)*. *Requirements Engineering* 3(2), S. 73-78, 1998.
- [6] H. Kilov, B. Rumpe: *Second ECOOP Workshop on Precise Behavioral Semantics (with an Emphasis on OO Business Specifications)*. In S. Demeyer, J. Bosch, *Object-Oriented Technology: ECOOP'98 Workshop Reader*, S. 167-188, Springer, 1999.
- [7] A. Evans, R. France, K. Lano, B. Rumpe: *The UML as a Formal Modeling Notation*. In J. Bézivin, P.-A. Muller, *The Unified Modeling Language*,

- UML'98: Beyond the Notation, First International Workshop, Selected Papers*, S. 336-348, Springer, 1998.
- [8] S. Kent, A. Evans, B. Rumpe: *UML Semantics FAQ*. In A. Moreira, S. Demeyer, *Object-Oriented Technology: ECOOP'99 Workshop Reader*, LNCS 1743, Springer, S. 33-56, 1999.
 - [9] K. Lano, J. Bicarregui: *Semantics and Transformations for UML Models*. In J. Bézivin and P.-A. Muller, *The Unified Modeling Language, UML'98: Beyond the Notation, First International Workshop, Selected Papers*, S. 107-119, Springer, 1998.
 - [10] J.C. Bicarregui, K.C. Lano, T.S.E. Maibaum: *Formalising Object-Oriented Models in the Object Calculus*. In J. Bosch, S. Mitchell, *Object-Oriented Technology: ECOOP'97 Workshop Reader*, S. 155-160, Springer, 1998.
 - [11] H.-D. Ehrich: *Object Specification*. In E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner, *Algebraic Foundations of Systems Specification*, S. 435-465, Springer, 1998.
 - [12] P. Zave, M. Jackson: *Four Dark Corners of Requirements Engineering*. IEEE Transactions on Software Engineering and Methodology, 6(1), S. 1-30, 1997.
 - [13] D. L. Parnas, D. K. Peters: *An Easily Extensible Toolset for Tabular Mathematical Expressions*. In W. R. Cleaveland, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS'99*, S. 345-359, Springer, 1999.
 - [14] C. Heitmeyer, J. Kirby, B. Labaw, R. Bharadway: *SCR*: A Toolset for Specifying and Analyzing Software Requirements*. In A. J. Hu, M. Y. Vardi, *Proceedings of 10th International Conference Computer Aided Verification CAV'98*, S. 526-531, Springer, 1998.
 - [15] N. Medvidovic, R. N. Taylor: *A Framework for Classifying and Comparing Architecture Description Languages*. In M. Jazayeri, H. Schauer, *Software Engineering - ESEC/FSE '97*, S. 60-76, Springer, 1997.
 - [16] G. Goos, W. Zimmermann: *Verifying Compilers and ASMs*. Erscheint in *International Workshop on Abstract State Machines ASM'2000*, LNCS, Springer, 2000.
 - [17] E. Börger: *Composition and Structuring Principles for ASMs*. Erscheint in *International Workshop on Abstract State Machines ASM'2000*, LNCS, Springer, 2000.
 - [18] Y. Gurevich, D. Rosenzweig: *Partially Ordered Runs: A Case Study*. Erscheint in *International Workshop on Abstract State Machines ASM'2000*, LNCS, Springer, 2000.

3.4 Eigene Vorarbeiten

Die eigenen Vorarbeiten bezüglich der Arbeitsschwerpunkte A und B sind im Arbeits- und Ergebnisbericht aufgeführt. Hinsichtlich des Arbeitsschwerpunkts C wurden folgende vorbereitenden Arbeiten durchgeführt. In [19] wird ein verteilter Algorithmus von Dijkstra, Feijen und van Gasteren auf zwei unterschiedlichen Abstraktionsebenen spezifiziert und mit Hilfe einer Verfeinerungsrelation verifiziert. In [20] wird ein Einblick in die auf ASMs basierende formale Semantik von SDL-2000 gegeben, welche zur Zeit innerhalb eines internationalen Projekts von einem Kreis von Experten entwickelt wird. SDL ist eine von der ITU (International Telecommunication Union) standardisierte Sprache und ist durch die ITU-T (Telecommunication Standardization Sector der ITU) definiert. Das Semantik-Projekt wird koordiniert von R. Gotzhein (Leiter Teilprojekt B4), der innerhalb der ITU-T die Funktion "Associate Rapporteur: Formal Semantics for SDL-2000" inne hat.

- [19] R. Eschbach: *A Termination Detection Algorithm: Specification and Verification*. In J. M. Wing, J. Woodcock, J. Davies, *Proc. of FM'99 - World Congress on Formal Methods in the Development of Computing Systems*, LNCS 1709, Springer, S. 1720-1737, 1999.
- [20] R. Eschbach, U. Glässer, R. Gotzhein, A. Prinz: *On the Formal Semantics of SDL-2000: A Compilation Approach using Abstract State Machines*. Erscheint in *International Workshop on Abstract State Machines ASM2000*, LNCS, Springer, 2000.

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Ziel des Teilprojekts C1 ist weiterhin die Koordinierung und Integration formaler Beschreibungstechniken, die im gesamten SFB verwendet werden. Ein langfristiges Ziel dabei ist die Entwicklung eines Referenzmodells für die Software-basierte Entwicklung großer, reaktiver Systeme.

Unter einem System verstehen wir eine ablauffähige Einheit von Hard- und Softwarekomponenten zusammen mit der beeinflussenden und beeinflussten Umgebung dieser Komponenten. Unter einem reaktiven System verstehen wir ein System, bei dem es eine ständige Interaktion zwischen der Umgebung und den Hard- und Softwarekomponenten gibt.

Hiervon zu unterscheiden sind Beschreibungen von Systemen. Diese werden für die Entwicklung von Systemen benötigt, weil sie es erlauben, Systeme zu verstehen, zu analysieren und sich über sie zu verständigen. Dokumente sind Beschreibungen eines Systems, die mit Hilfe einer oder mehrerer Beschreibungstechniken erstellt werden und eine Struktur besitzen. Diesen Dokumenten liegt also eine bestimmte Syntax und Semantik zugrunde.

Mit Hilfe einer Beschreibung eines Systems lässt sich auch ein Maß für die Größe

eines Systems angeben. Ein solches Maß könnte beispielsweise aus der Anzahl der betrachteten Phänomene - zum Beispiel die Helligkeit in einem Raum oder der übermittelte Wert eines Temperatursensors - und der Anzahl der beschriebenen Eigenschaften dieser Phänomene bestehen.

Da also Dokumente von zentraler Bedeutung bei der Entwicklung von Systemen sind, gehen wir bei der Entwicklung des langfristig angestrebten Referenzmodells wie folgt vor. Es werden präzise Modelle der Inhalte von Dokumenten erstellt, geeignete Beschreibungstechniken zur Repräsentation dieser Inhalte untersucht, kombiniert und entwickelt sowie Methoden und Vorgehensweisen zur Erstellung solcher Dokumente erforscht und verbessert. Konkrete Ziele dieser Arbeiten sind, das Erstellen formaler Beschreibungen, unter besonderer Berücksichtigung der Konzepte Wiederverwendung und Komposition, zu unterstützen sowie die vielfältigen Arten der semantischen Beziehungen zwischen den Dokumenten mit Hilfe geeigneter Verfolgbarkeitsrelationen darzustellen.

Diese Vorgehensweise hat sich bei der Erstellung eines Referenzmodells im Bereich der Anforderungen bewährt. Dieses Referenzmodell legt präzise den Inhalt eines Dokuments, das wir Problemspezifikation nennen, fest. Hierzu wird ein System unter dem Aspekt *Vorhandensein eines Phänomens* aufgespalten in den Teil, der bereits vorhanden ist, die *Umwelt*, und den Teil, der noch zu entwickeln ist, die *Maschine*.

Entsprechend werden die Eigenschaften eines Systems aufgeteilt in indikative und optative Eigenschaften. *Indikative* Eigenschaften besitzt ein System ohne bzw. trotz einer Maschine. Sie beschreiben Fakten und bilden das sogenannte *Domänenwissen*. Das Verhalten bereits installierter Sensoren, zum Beispiel eines Bewegungsmelders, der eine 1 liefert, wenn sich eine Person in einem Raum bewegt, ist beispielsweise eine Eigenschaft eines Domänenwissens. *Optative* Eigenschaften hingegen beschreiben Wünsche an das System und müssen durch eine noch zu entwickelnde Maschine sichergestellt werden. Die Menge der optativen Eigenschaften wird weiter unterteilt in *Anforderungsspezifikation* und *Maschinenspezifikation*.

Während eine in einer Maschinenspezifikation enthaltene Eigenschaft prinzipiell realisierbar ist, weil alle in ihr verwendeten Phänomene für die Maschine *sichtbar* sind, kann dies bei einer Eigenschaft in einer Anforderungsspezifikation nicht garantiert werden. Im Bereich der Gebäudeautomatisierung ist zum Beispiel die Aussage „Wenn eine Person in einem Raum ist, soll das Licht an sein.“ eine typische Eigenschaft in einer Anforderungsspezifikation. Die entsprechende Eigenschaft in einer Maschinenspezifikation könnte lauten: „Wenn ein Bewegungsmelder in einem Raum eine 1 liefert, muss der Lichtschalter auf Ein stehen.“ Wenn die Sensoren und Aktuatoren bereits installiert sind, kann diese Eigenschaft prinzipiell, d.h. von der Sichtbarkeit her, realisiert werden.

Um die Eignung dieses Referenzmodells zu überprüfen, wurde es für große, reaktive Systeme instanziiert. Hierzu wurde unter anderem eine Realzeit-Temporallogik mit objektorientierten Konzepten wie Vererbung und Aggregation sowie der Möglichkeit

zu parametrisieren erweitert. Damit ist es möglich, im Bereich großer, reaktiver Systeme die oben beschriebenen Informationen in einer Problemspezifikation adäquat zu repräsentieren und die Wiederverwendung einzelner Teile einer Problemspezifikation zu unterstützen. Zur Unterstützung der Erstellung von Problemspezifikationen wurde zum einen ein auf Anforderungsmustern basierendes, wiederverwendungsorientiertes Prozessmodell entwickelt und zum anderen ein Werkzeug realisiert. Weitere Details zu diesen Arbeiten sind im Ergebnisbericht dokumentiert.

Im Arbeitsschwerpunkt A stehen Problemspezifikationen im Mittelpunkt der Untersuchungen. Dabei wird die Eignung des bisher entwickelten Referenzmodells und seiner Instanziierung weiter überprüft sowie die methodische Unterstützung bei der Erstellung und Verwendung von Problemspezifikationen verbessert. Die Überprüfung der Eignung erfolgt dabei anhand weiterer Fallstudien, wobei zunehmend auch andere Anwendungsdomänen reaktiver Systeme, wie zum Beispiel die bereits in der zweiten Förderungsperiode betrachteten Verkehrsbeeinflussungsanlagen, betrachtet werden. Zur Verbesserung der methodischen Unterstützung werden aufbauend auf den bisher gemachten Erfahrungen bei der Erstellung von Problemspezifikationen Kompositionsooperatoren definiert, die insbesondere auch semantische Eigenschaften eines resultierenden Produkts garantieren. Neben diesem mehr generierenden Konzept werden auch Konzepte und Verfahren entwickelt, die eine Analyse semantischer Eigenschaften einer gegebenen Problemspezifikation ermöglichen. Hierbei spielt die horizontale Verfolgbarkeit innerhalb einer Problemspezifikation eine zentrale Rolle. Weiterhin ist es bei der Verwendung von Problemspezifikationen großer Systeme unbedingt erforderlich, verschiedene, zueinander konsistente Projektionen einer Problemspezifikation erstellen zu können. Auch dabei werden verschiedene horizontale Verfolgbarkeitsrelationen benötigt. Bei all diesen Ansätzen und Konzepten zur Verbesserung der methodischen Unterstützung der Erstellung und Verwendung von Problemspezifikationen ist stets deren Integration in das existierende Werkzeug ein wesentliches Ziel.

Der Arbeitsschwerpunkt B beschäftigt sich mit Entwurfsspezifikationen. Dabei wird zunächst das vorhandene Referenzmodell durch Einbeziehung von Entwurfsspezifikationen erweitert. Hierzu muss geklärt werden, welche Inhalte unabhängig von einer bestimmten Beschreibungstechnik in einem solchen Dokument abzulegen sind. Neben dem Aspekt des Verhaltens ist hier auch der Aspekt der Architektur der zu entwickelnden Maschine von zentraler Bedeutung. Analog zu Problemspezifikationen wird anschließend untersucht, welche Beschreibungstechniken geeignet sind, diese Inhalte darzustellen. Die Kombination verschiedener Beschreibungstechniken, vor allem solcher, die das Verhalten operational beschreiben, und solcher, die die Architektur spezifizieren, ist auch hier unumgänglich. Dabei muss insbesondere berücksichtigt werden, dass nicht nur Architekturen einzelner Systeme beschrieben werden, sondern insbesondere auch konfigurierbare Architekturen ganzer Systemfamilien. Geeignete Beschreibungstechniken müssen also auch Möglichkeiten zur Parametrisierung struktureller Eigenschaften zur Verfügung stellen. Beispielsweise werden in Abhängigkeit der betrachteten Aspekte, wie zum Beispiel Temperatur, Helligkeit oder

Sicherheit, unterschiedliche Architekturen notwendig sein, um die verschiedenen sich ergebenden Konfliktsituationen adäquat auflösen zu können. Für solche strukturellen Kombinationen werden langfristig auch geeignete Kompositionsoperatoren entwickelt. Von zentraler Bedeutung ist außerdem auch hier wieder das Problem der Verfolgbarkeit, diesmal zunächst vertikal zwischen Problemspezifikationen und Entwurfsspezifikationen. Dabei wird auch geklärt, inwieweit sich die für Problemspezifikationen entwickelten Kompositionsoperatoren auf Entwurfsspezifikationen übertragen lassen.

In einem sehr engen Zusammenhang hierzu stehen die Arbeiten im Arbeitsschwerpunkt C, wo zum einen semantische Beziehungen zwischen temporallogischen Problemspezifikationen und darauf aufbauenden Entwurfsspezifikationen und zum anderen Verfeinerungsrelationen zwischen Entwurfsspezifikationen untersucht werden.

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Problemspezifikationen

Bei den Arbeiten dieses und des folgenden Arbeitsschwerpunktes lassen wir uns im Wesentlichen von den Erfahrungen leiten, die wir beim Erstellen und Verwenden von Problemspezifikationen im Bereich der Gebäudeautomatisierung (zum Beispiel Praktika, Teams) und in anderen Anwendungsfeldern (zum Beispiel Verkehrsbeeinflussungsanlagen) machen bzw. bereits gemacht haben. Alle hierbei entwickelten Konzepte und Ansätze werden in das existierende Werkzeug integriert. Bei allen Arbeiten müssen die Hierarchien berücksichtigt werden, die durch die in unserem Ansatz verwendeten Konzepte Aggregation und Vererbung in einer Problemspezifikation vorliegen.

Arbeitspaket A1: Validation des Referenzmodells

Die Eignung des Referenzmodells für Problemspezifikationen wird anhand weiterer Fallstudien überprüft. Dabei sollen zunehmend auch andere Anwendungsdomänen betrachtet werden. Eine zentrale Frage hierbei ist, ob die bisher zur Spezifikation verwendete Temporallogik ausreichend ist, um alle Eigenschaften adäquat zu erfassen, oder ob weitere Beschreibungstechniken, wie zum Beispiel Tabellen oder graphische Notationen, integriert werden müssen. Dabei muss stets sichergestellt werden, dass jede einzelne Beschreibungstechnik wie auch deren Kombination nicht nur eine präzise Syntax, sondern auch eine mathematisch fundierte Semantik besitzen.

Personalbedarf: 5 PM

Arbeitspaket A2: Kompositionsoperatoren für Problemspezifikationen

Zur Verbesserung des Erstellungsprozesses werden Kompositionsoperatoren integriert. Hierfür wird eine Sammlung von Kompositionsoperatoren erstellt. Für deren

jeweilige Verwendungsmöglichkeit werden entsprechende Richtlinien aufgestellt. Kompositionsoperatoren sind von Nutzen, sowohl wenn semantische Eigenschaften, wie zum Beispiel der Zusammenhang zwischen Domänenwissen, Maschinenspezifikation und Anforderungsspezifikation, als auch wenn mehr syntaktische Eigenschaften, wie zum Beispiel die Vollständig- und Eindeutigkeit einer Fallunterscheidung, betrachtet werden. Eine Integration der entwickelten Kompositionsoperatoren in unser Werkzeug ist unbedingt erforderlich, um es für die Entwicklung großer Systeme effektiver einsetzen zu können.

Personalbedarf: 9 PM

Arbeitspaket A3: Analyse von Problemspezifikationen

Bisher garantiert das Werkzeug aufgrund einiger syntaktischer Randbedingungen eine gewisse (syntaktische) Konsistenz einer Problembeschreibung, wie zum Beispiel eine zyklusfreie Klassenhierarchie. In diesem Bereich ist noch eine große Anzahl weiterer syntaktischer Überprüfungen möglich, beispielsweise ob zumindest jede maschineninterne Größe (zum Beispiel eine Funktion) einen Initialwert besitzt oder ob eine Größe aufgrund einer Problemspezifikation zu jedem Zeitpunkt genau einen Wert besitzt (d.h. eine Fallunterscheidung muss eindeutig und vollständig sein). Solche Überprüfungen decken zum einen Lücken in einer Problemspezifikation auf, zum anderen erhöhen sie beim Benutzer eines solchen Dokuments das Vertrauen in dieses. Im Zusammenhang mit Arbeitspaket A2 muss geklärt werden, welche Eigenschaften bereits durch die Verwendung von Kompositionsoperatoren garantiert werden können und welche explizit zu überprüfen sind.

Im Unterschied zu Arbeitspaket A5 wird hier stets nur eine Art der Beschreibung (Domänenwissen, Maschinenspezifikation oder Anforderungsspezifikation) betrachtet und nicht alle zusammen.

Personalbedarf: 6 PM

Arbeitspaket A4: Projektionen und Verfolgbarkeit

Die Verwendung der bisher erstellten Problemspezifikationen hat gezeigt, dass deren Struktur sowie die verschiedenen Arten von Navigationshilfen (insbesondere im WWW) das Auffinden von Informationen und damit die Verständlichkeit des Gesamtdokumentes unterstützen. Häufig ist es jedoch wünschenswert, einen eingeschränkten Auszug einer Problemspezifikation zur Verfügung zu haben, um sich über einen bestimmten Teil einer Problemspezifikation leichter und schneller einen Überblick verschaffen zu können. Beispielsweise ist es oft ausreichend, nur den Teil einer Problemspezifikation zu betrachten, der sich mit einem bestimmten Aspekt wie Temperatur oder Helligkeit beschäftigt. Hierbei muss nicht nur garantiert werden, dass verschiedene Projektionen konsistent zueinander sind, sondern es muss auch untersucht werden, was geeignete Projektionen sind und wie sie erstellt werden können, zum Beispiel welche Hierarchien wie aufgelöst werden sollen. Hier spielen natürlich auch verschiedene Ausprägungen von Verfolgbarkeitsrelationen eine zentrale Rolle.

Personalbedarf: 8 PM

Arbeitspaket A5: Beweisaufgaben und Verfolgbarkeit

Schließlich werden in diesem Arbeitsschwerpunkt Ansätze entwickelt, die die Lösung typischer Beweisaufgaben unterstützen, die beim Erstellen von Problemspezifikationen anfallen. Ein Beispiel für eine in Problemspezifikationen immer wiederkehrende Beweisaufgabe ist der Nachweis, dass das Domänenwissen zusammen mit der Maschinenspezifikation die Anforderungsspezifikation erfüllt. Diese Beweise sind in der Regel lokal begrenzt, d.h. es muss immer nur eine kleine Teilmenge der Eigenschaften betrachtet werden. Hier ist eine Frage, inwieweit automatisch festgestellt werden kann, welche Eigenschaften für einen Beweis berücksichtigt werden müssen, d.h. welche Eigenschaften Seiteneffekte haben können. Hierbei kommt eine neue Art der Verfolgbarkeit ins Spiel, nämlich zwischen Eigenschaften der verschiedenen Beschreibungsarten (Domänenwissen, Anforderungsspezifikation, Maschinenspezifikation). In diesem Zusammenhang wird auch ansatzweise untersucht, welche existierenden Beweiser bei der Bewältigung der anfallenden Beweisaufgaben wie eingesetzt werden können. Des Weiteren treten bei der Verwendung von Kompositionsoperatoren (siehe Arbeitspaket A2) Bedingungen auf, die für die zusammenzusetzenden Komponenten nachzuweisen sind.

Personalbedarf: 8 PM

Arbeitsschwerpunkt B: Entwurfsspezifikationen

Ziel dieses Arbeitsschwerpunktes ist es, das bisher entwickelte Referenzmodell auf Entwurfsspezifikationen zu erweitern. Neben der Festlegung des Inhalts eines solchen Dokuments und der Untersuchung geeigneter Beschreibungstechniken zur Repräsentation dieses Inhalts besteht ein wesentliches Ziel bei diesen Arbeiten in der Aufstellung geeigneter Verfolgbarkeitsrelationen zwischen Problemspezifikationen und Entwurfsspezifikationen.

Die Eignung der in diesem Arbeitsschwerpunkt entwickelten Modelle, Konzepte und Verfahren wird anhand von Fallstudien überprüft, wobei wir auf die bereits vorhandenen Problemspezifikationen aus den Bereichen der Gebäudeautomatisierung und der Verkehrsbeeinflussungsanlagen aufbauen.

Arbeitspaket B1: Entwicklung eines Referenzmodells

Zunächst ist zu untersuchen, welche Inhalte in einer Entwurfsspezifikation abgelegt werden sollen, d.h. es muss analog zu den Problemspezifikationen ein geeignetes Referenzmodell für Entwurfsspezifikationen erstellt werden. Beispielsweise ist es in einer Entwurfsspezifikation sicherlich nur noch erforderlich, Größen zu verwenden, die für die Maschine sichtbar sind. Weiterhin wird in einer Entwurfsspezifikation auch die Architektur der zu entwickelnden Maschine festgelegt. Insgesamt ist eine Entwurfsspezifikation lösungsorientiert, wohingegen eine Problemspezifikation problemorientiert ist.

Ausgangspunkt dieser Arbeiten sind die in den vergangenen beiden Förderungsperioden, insbesondere in den Querschnittsprojekten, gemachten Erfahrungen.

Personalbedarf: 10 PM

Arbeitspaket B2: Beschreibungstechniken für Entwurfsspezifikationen

Die inhaltlichen Besonderheiten von Entwurfsspezifikationen im Vergleich zu Problemspezifikationen werden sich auch in der Wahl geeigneter (formaler) Beschreibungstechniken niederschlagen. Während zur Spezifikation eines Problems deskriptive Beschreibungstechniken wie Logiken als geeignet angesehen werden, wird zur Beschreibung einer Lösung eine mehr operationale Beschreibungstechnik oder eine Beschreibungstechnik, die beide Attribute in sich vereinigt, benötigt.

Desweiteren wird bei der Bewertung und Auswahl von Beschreibungstechniken für Entwurfsspezifikationen berücksichtigt, dass in einer Entwurfsspezifikation neben dem Verhalten auch die Architektur der zu entwickelnden Maschine beschrieben werden muss. Bezüglich des Verhaltens werden die formalen Beschreibungstechniken SDL und ASM (Abstract State Machines) als Ausgangspunkt gewählt. Hinsichtlich der Architekturbeschreibungssprachen erfolgt ausgehend von [15] zunächst noch eine Sichtung und Bewertung existierender Techniken.

Personalbedarf: 10 PM

Arbeitspaket B3: Verfolgbarkeit zwischen Problem- und Entwurfsspezifikationen

Ein wesentlicher Aspekt bei der Erstellung eines Referenzmodells für Entwurfsspezifikationen ist dessen Koppelung an das Referenzmodell für Problemspezifikationen. Dies bedeutet vor allem, dass eine Verfolgbarkeitsrelation zwischen einer Problemspezifikation und einer darauf aufbauenden Entwurfsspezifikation festgelegt wird, die sowohl die einzelnen Bezeichner, die Eigenschaften als auch die Struktur berücksichtigt, die in einer Problemspezifikation vorgegeben sind.

Personalbedarf: 6 PM

Arbeitspaket B4: Kompositionsoperatoren für Entwurfsspezifikationen

Ebenso wie bei Problemspezifikationen sollte auch die Erstellung von Entwurfsspezifikationen mit Hilfe geeigneter Kompositionsoperatoren unterstützt werden. Hier ist insbesondere zu untersuchen, inwieweit die Kompositionsoperatoren, die bei der Erstellung von Problemspezifikationen verwendet werden, auch zur Erstellung von Entwurfsspezifikationen verwendet werden können; denn bei Problemspezifikationen steht nahezu ausschließlich die Komposition des Verhaltens im Mittelpunkt, während bei Entwurfsspezifikationen zumindest auch noch strukturelle Aspekte eine wesentliche Rolle spielen.

Personalbedarf: 10 PM

Arbeitsschwerpunkt C: Verfeinerung von ASM-Entwurfsspezifikationen

Ziel dieses Arbeitsschwerpunktes ist es, semantische Beziehungen zwischen Problemspezifikationen und den zugeordneten Entwurfsspezifikationen bzw. Verfeinerungsrelationen zwischen Entwurfsspezifikationen zu untersuchen. Dies soll vor allem anhand der formalen Beschreibungstechniken ASM (Abstract State Machines) und SDL (Specification and Description Language) geschehen. Des Weiteren sollen

Strukturierungsmethoden für ASM-Entwurfsspezifikationen entwickelt und untersucht werden.

Arbeitspaket C1: Erfüllbarkeit zwischen Problem- und Entwurfsspezifikationen

Beim Übergang einer Problemspezifikation zu einer Entwurfsspezifikation muss garantiert werden, dass die Eigenschaften der Problemspezifikation von der Entwurfsspezifikation erfüllt werden. In diesem Arbeitspaket soll anhand von ASM untersucht werden, wie Erfüllbarkeit zwischen Entwurfsspezifikationen und Problemspezifikationen definiert werden kann. Was bedeutet also, dass eine Entwurfsspezifikation eine Problemspezifikation erfüllt? Diese Frage soll mathematisch exakt beantwortet werden.

Personalbedarf: 4 PM

Arbeitspaket C2: Verfeinerung von Entwurfsspezifikationen

Entwurfsspezifikationen werden innerhalb des Entwicklungsprozesses verändert, etwa wenn eine abstrakte Spezifikation “konkretisiert” wird, wenn also zum Beispiel weitere Implementationsdetails der Spezifikation hinzugefügt werden. Dieser Vorgang wird im Allgemeinen Verfeinerung genannt und soll in diesem Arbeitspaket genauer untersucht werden. Wie können also Verfeinerungsrelationen zwischen Entwurfsspezifikationen aussehen? Welche sinnvollen Anforderungen an diese Relationen gibt es? Eine mögliche Anforderung ist zum Beispiel die Forderung, dass Verfeinerungsrelationen die Erfüllbarkeitsrelation(en) aus Arbeitspaket C.1 erhalten sollen. Wenn also eine Entwurfsspezifikation eine Problemspezifikation erfüllt, dann soll dies auch jede Verfeinerung der Entwurfsspezifikation tun.

Personalbedarf: 14 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausrüstung R. Eschbach						
	C1 und C2					
Ergänzungsausrüstung M. Kronenburg						
	A1					
	A2					
	A3					
			A4			
			A5			
	N.N.	B1				
		B2				
		B3				
					B4	
N.N., stud. Hilfskraft						
	A1					
	A2					
			A4			
N.N., stud. Hilfskraft						
	A1					
	A3					
			A5			

Im Zeitplan ist ein Aufwand von wenigstens 8 Std. / Woche für SFB-übergreifende Themenstellungen berücksichtigt, die jedes Teilprojekt zur Verfügung stellt. Diese Arbeiten werden in Teams organisiert und im Arbeitsbericht gesondert beschrieben.

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Als Infrastrukturprojekt arbeitet dieses Teilprojekt mit allen anderen Teilprojekten zusammen. Es unterstützt diese bei der Modellierung und koordiniert die Verwendung von Modellen und Beschreibungstechniken.

Aufbauend auf der erfolgreichen Zusammenarbeit mit den Teilprojekten B10, D1 und D2 im Querschnittsprojekt Q3 sind insbesondere mit diesen Teilprojekten sowie dem neuen Teilprojekt B11, dessen Leiter am Querschnittsprojekt Q3 beteiligt ist, enge Kooperationen geplant. Dabei werden wir B10 bei der Formalisierung des Design Space-Konzeptes unterstützen. Mit Teilprojekt B11 ist insbesondere eine enge

Absprache vorgesehen hinsichtlich der Untersuchungen im Bereich der Entwurfsphase. Teilprojekt D1 werden wir bei der Formalisierung der Systematik beratend unterstützen. Bezüglich der Arbeiten zur Erfassung von Anforderungen ergänzen sich unsere Aktivitäten und die von Teilprojekt D1. Teilprojekt D2 werden wir bei der Suche nach einer geeigneten formalen Repräsentation ihrer mathematischen Modelle und Algorithmen unterstützen. Da erste Gespräche gezeigt haben, dass hierfür voraussichtlich eine mehr deskriptive Beschreibung und auf objektorientierten Konzepten basierende Strukturierung geeignet ist, sind unsere diesbezüglichen Erfahrungen hier sehr hilfreich. Da im gesamten SFB die formale Beschreibungstechnik SDL verwendet wird, ist die unter 3.4 beschriebene Mitarbeit an der formalen Semantik von SDL für den SFB von Nutzen.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	1	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
			StdHkr.	2	40.800	StdHkr.	2	40.800	StdHkr.	2	40.800
	zus.:	1	zus.:	4	249.600	zus.:	4	249.600	zus.:	4	249.600
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		3.000	(522)		3.000	(522)		3.000
			zusammen		3.000	zusammen		3.000	zusammen		3.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausrüstung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Prof. Dr. K. Madlener	Informatik	FB Inf., Uni KL	5	1.1.95	
	(2) Prof. Dr. J. Avenhaus	Informatik	FB Inf., Uni KL	5	1.1.95	
	(3) Dipl. Inform. R. Eschbach	Informatik	FB Inf., Uni KL	20	1.1.98	
Ergänzungsausrüstung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Dipl-Inform. M. Kronenburg	Informatik	FB Inf., Uni KL	38,5		IIa
	(2) N.N., Dipl-Inform.	Informatik	FB Inf., Uni KL	38,5		IIa
	(3) N.N., StdHkr.	Informatik	FB Inf., Uni KL	19		
	(4) N.N., StdHkr.	Informatik	FB Inf., Uni KL	19		
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Der wissenschaftliche Mitarbeiter ist für die Bearbeitung des Arbeitsschwerpunktes C vorgesehen. Da in diesem Arbeitsschwerpunkt Entwurfsspezifikationen in einer konkreten Beschreibungstechnik, nämlich ASM, betrachtet werden, ist es erforderlich, dass zwischen diesem Mitarbeiter und den beiden Mitarbeitern der Ergänzungsausrüstung eine enge Kooperation stattfindet.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Der erste wissenschaftliche Mitarbeiter ist für die Bearbeitung des Arbeitsschwerpunktes A vorgesehen. Neben der Erarbeitung neuer Konzepte, die die Erstellung und Analyse von Problemspezifikationen unterstützen, steht hierbei vor allem deren Integration in das existierende Werkzeug im Vordergrund.

Der zweite wissenschaftliche Mitarbeiter ist für die Bearbeitung des Arbeitsschwerpunktes B vorgesehen. Hierbei handelt es sich um schwierige und anspruchsvolle Arbeiten auf konzeptioneller Ebene.

Die beantragten studentischen Hilfskräfte werden vor allem zur Unterstützung der Arbeiten im Arbeitsschwerpunkt A benötigt, da hier zum einen umfangreiche Fallstudien (Arbeitspaket A1) durchgeführt werden sollen und zum anderen ein großer Implementierungsaufwand anfällt (Arbeitspakete A2-A5).

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausrüstung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausrüstung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausrüstung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	3.000	3.000	3.000

Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

Die sächlichen Verwaltungskosten in der Ergänzungsausrüstung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
	-	-	-
Summe	-	-	-

Projektbereich D

Prototypanwendungen

3 Prototypanwendungen

3.01 Zusammenfassende Darstellung des Projektbereiches

Der Projektbereich D besteht aus den beiden Teilprojekten D1 und D2 der Fachbereiche Informatik und Elektrotechnik. Im Vordergrund stehen die physikalisch und technisch relevanten Aspekte der Anwendungsdomäne *Gebäudeautomation*. Einerseits werden Infrastrukturaufgaben aus dem Anwendungsgebiet bearbeitet, andererseits werden neue Steuerungs- und Regelungskonzepte erforscht. Gerade in diesem Projektbereich ist die interdisziplinäre Zusammenarbeit und Ergänzung von besonderer Bedeutung.

Geeignete Aufgabenstellungen für Experimente zur Entwicklung großer Systeme werden zur Verfügung gestellt. Gleiches gilt für Testumgebungen der Produkte in allen Stadien der Entwicklung. Aufgrund der Wahl des Anwendungsbereiches *reaktive eingebettete Systeme* ist das Testen in realen oder simulierten Umgebungen unabdingbar. Nur so kann die Qualität der Produkte bezüglich Vollständigkeit, funktionaler Korrektheit und der Einhaltung nichtfunktionaler Anforderungen beurteilt werden.

Die Anwendung *Gebäudeautomation* erfordert in besonderem Maße physikalisch technisches Domänenwissen. Dies umfasst Wissen bezüglich der physikalischen Zusammenhänge von Heizung, Klima, Lüftung und der Lichttechnik. Aber auch sicherheitsrelevante Betrachtungen, gesetzliche Normen und Vorschriften sind von Bedeutung. Dieses Wissen wird vom Projektbereich D erarbeitet und zur Verfügung gestellt. Dies kann direkt oder nach geeigneter Aufbereitung mittels der Erfahrungsdatenbank geschehen.

Besonders in der Analysephase des Entwicklungsprozesses großer Automatisierungssysteme sind neue Strategien für die Automatisierung zu entwickeln. So werden neue Konzepte modellbasierter, validierter Steuerungs- und Regelungsalgorithmen entworfen. Neben den simulativen Tests werden Validierungen in realen Testumgebungen erfolgen.

3.02 Bisherige und beantragte Förderung des Projektbereiches

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	681,6	40,0	55,0	776,6
1998	403,2	20,0	-	423,2
1999	412,8	29,0	-	441,8
2000	417,6	17,4	-	435,0
Zwischensumme	1915,2	106,4	55,0	2076,6
2001	499,2	39,0	-	538,2
2002	499,2	34,0	-	533,2
2003	499,2	32,0	-	531,2

(jeweils Ergänzungsausstattung, Beträge in DM 1000,-)

3.1 Allgemeine Angaben zum Teilprojekt D1

3.1.1 Thema

Anwendungssystem Gebäude

3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, Modellierung, Reaktive Systeme, Gebäudeautomation

3.1.3 Leiter

Prof. Dr. Gerhard Zimmermann
geb. 18.05.1939
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2628
Fax: 0631/205-2162
zimmerma@informatik.uni-kl.de

Die Stelle des Leiters ist befristet: ☒ (x) nein ☐ () ja

3.1.4 -entfällt-

3.1.5 In dem Teilprojekt sind vorgesehen

Untersuchungen am Menschen	☐ () ja	☒ (x) nein
klinische Studien im Bereich		
der somatischen Zell- oder Gentherapie	☐ () ja	☒ (x) nein
Tierversuche	☐ () ja	☒ (x) nein
gentechnologische Untersuchungen	☐ () ja	☒ (x) nein

3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Haushaltsjahr	Personalkosten bzw. BAT-Stellen	Sächliche Verwal- tungsausgaben	Investitionen	Gesamt
bis 1997	681,6	40	55	776,6
1998	201,6	20	-	221,6
1999	206,4	20	-	226,4
2000	208,8	5	-	213,8
Zwischensumme	1298,4	45	55	1438,4
2001	249,6	20	-	269,6
2002	249,6	15	-	264,6
2003	249,6	10	-	259,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000,-)

3.2 Zusammenfassung

Das Teilprojekt D1 stellt als Hauptaufgabe die Infrastruktur für Experimente in der Anwendungsdomäne *Gebäudeautomation* zur Verfügung und sucht darüberhinaus domänenspezifische Lösungen für die Requirements-Engineering-Phase der Entwicklung großer, reaktiver Systeme. Aus beiden Teilen resultieren Entwicklungs- und Forschungsaufgaben.

Die Qualität der in den Teilprojekten des SFB entstehenden Entwicklungsmethoden und -werkzeuge lässt sich nur mit genügend großen, realistischen Experimenten in einer kontrollierbaren *Testumgebung* messen und bewerten. Dazu stellt D1 Anwendungsszenarien, ein konfigurierbares *Testfeld* mit Sensoren, Aktuatoren, lokalen Netzen und Prozessoren, *Simulatoren* und eine *Prototyping-Umgebung* zur Verfügung. D1 unterstützt die Teilprojekte und die projektübergreifenden Teams bei der Anpassung und Benutzung der Testumgebung.

Wir haben eine ereignisgesteuerte, objektorientierte Simulationstechnik entwickelt, welche die bei reaktiven Systemen häufig auftretenden Sprungfunktionen (binäre Sensoren, stochastische Ereignisse) besonders gut mit kontinuierlichen Funktionen verknüpft. Darüberhinaus wurde eine Modellierungsmethode entwickelt, die es erlaubt, mit Hilfe domänenspezifischer Design Pattern weitgehend automatisch für jedes Experiment und jede Modifikation während eines Experiments Spezialsimulatoren zu generieren. In der nächsten Periode wollen wir zeigen, wie effizient diese *Simulator-Entwicklungsmethode* in enger Kopplung mit dem Analyseprozess ist, welche Simulatorarchitekturen geeignet sind und wie Simulatoren in das Testfeld integriert werden können, um dieses skalierbar zu machen.

Die technischen Voraussetzungen für das *Prototyping* während des Requirements-Engineering haben wir geschaffen. Prototypen werden aus den System Requirements über SDL-Modelle automatisch erzeugt und über die Prototyping-Umgebung mit dem Testfeld gekoppelt. In der nächsten Periode sollen Probleme des Prototyping während der Design-Phase gelöst werden. Dazu gehören Fragen des Prototyping von Teilmodellen und HW/SW-Systemen und das Partitionieren von Prototypen. Ein weiterer Schwerpunkt ist die Frage, wie mit Prototypen Aussagen über konkurrentes Verhalten und über nichtfunktionale Eigenschaften gemacht werden können.

Requirements-Engineering ist eine kritische Phase bei der Entwicklung reaktiver Systeme, die aber aus Kostengründen selten strikt durchgeführt wird. In einer Reihe von Fallstudien haben wir eine Modellarchitektur und einen Analyseprozess entwickelt und getestet, der in starkem Maße Reuse unterstützt und auf die Anwendungsdomäne zugeschnitten ist. Das neue Ziel ist die Entwicklung einer Entwurfssystematik für diesen Teilprozess, um alle Artefakte, Operationen und Entwicklungszustände auf einer soliden Basis zu definieren und verknüpfen. Damit soll auch die Verknüpfung von projektspezifischen Anforderungen über Struktur und Inhalt des Domänenlexikons mit der Reuse-Bibliothek hergestellt werden. Die Behandlung nichtfunktionaler Eigenschaften in dieser Phase soll untersucht und mit der Entwurfssystematik verknüpft werden.

3.3 Stand der Forschung

Hier beschränken wir uns auf die Gebiete, die sich in der letzten Periode stark entwickelt haben oder die in der beantragten Periode neu in Angriff genommen werden sollen.

Pattern: Die Notation von Entwurfsideen in Design Pattern hat sich in den letzten Jahren weiter etabliert. Zahlreiche neue Pattern und Pattern-Sprachen wurden aufgestellt ([1], [2], [3]). Die Bedeutung von Pattern-Systemen, also zueinander passenden und miteinander wirkenden Pattern, steht dabei mehr im Mittelpunkt. Pattern werden zunehmend zum Entwurf von objektorientierten Systemen verwendet (z.B. UML). Eine formale Methodik und feste Pattern-Schnittstellen gibt es leider noch nicht. Erste kommerzielle Werkzeuge haben Pattern als Modellierungseinheiten integriert (Rational Rose, Framework Studio, Together) und Code-Generierung aus konkreten Pattern wird eingesetzt (Surveyor, OmniBuilder). Dabei werden vor allem kleine, in vielen Situationen anwendbare Pattern verwendet, deren Generierungspotential nicht so hoch ist wie bei domänenspezifischen Pattern (z.B. Simulationpattern). Domänenspezifische Pattern werden vom Teilprojekt B4 zur Spezifikation und Generierung von Kommunikationssystemen eingesetzt [4]. Ähnlich zu unserem Ansatz beschäftigen sich diese Arbeiten auch mit der Integration des Pattern-Ansatzes in einen Entwurfsprozeß und mit der Einbringung in die Erfahrungsdatenbank des Sonderforschungsbereiches [5].

Simulation: Ein deutlicher Trend bei der Modellierung von Simulatoren geht hin zu objektorientierten Simulationssprachen (z.B. SMILE, Modelica). Dabei beschreibt jedes Simulationsobjekt sein Verhalten und seine Schnittstelle. Die Simulationsobjekte werden bei der Modellierung mit passenden Parametern instanziiert und durch Kanäle miteinander verbunden. Eine ähnliche Idee verfolgt unsere Modellierungsmethode, wobei die Beschreibung der zu simulierenden Effekte nicht in fertige Objekte gekapselt ist, sondern durch die Anbindung von effektspezifischen Design-Pattern an passende Objekte angewendet wird. Die Ausführung der Simulationsmodelle erfolgt bei den meisten Simulatoren durch das Lösen großer Differentialgleichungssysteme, wobei jedes Modellobjekt seine sich beschreibenden Gleichungen in das Gleichungssystem einbringt (z.B. Dymola [6]). Da bei unseren Simulationsaufgaben häufig Sprungfunktionen auftreten (binäre Sensoren, stochastische Ereignisse) und zur geeigneten Kopplung mit realer Hardware und Kontrollsystemen echtzeitbasiert simuliert werden muss, laufen unsere Simulationsmodelle ereignisgesteuert ab.

Entwurfsbegleitende Simulation verschiedener physikalischer Effekte wird auch im SEMPER-Projekt der Carnegie Mellon Universität untersucht [7]. Dabei dient der SEMPER-Simulator nicht dem Testen eines Steuerungssystems, sondern wird zur Optimierung des Gebäudeverhaltens eingesetzt. SEMPER ist aus verschiedenen Modulen, wie zum Beispiel der Lichtsimulation Lumina [8][9], zusammengesetzt.

Prototyping: An der TU München wurde das REAR-Framework für Rapid Prototyping von eingebetteten System mit harten Echtzeitanforderungen entwickelt [10]. Hardware- und Softwareprototypen werden aus um Realzeit-Annotationen ergänzten SDL-Modellen generiert und entsprechend instrumentiert, um die Zeiteigenschaften des Systems zu testen.

Im NIMBUS-Projekt der University of Minnesota werden Prototypen aus RSML-Modellen (Statemate) generiert [11]. Dabei kann auch die Hardware durch Modelle beschrieben werden.

Requirements-Engineering: Die Modellierung von SW- und HW/SW-Systemen ist in den letzten Jahren sehr stark in den Vordergrund gerückt, was die große Zahl internationaler Publikationen belegt. Stellvertretend soll hier nur der Vergleich [12] genannt werden. Speziell zu reaktiven Systemen gibt es eine steigende Zahl von Beiträgen, zum Beispiel zur Beschreibung von Echtzeitsystemen mit Hilfe von UML [13]. In Deutschland beschäftigen sich viele Gruppen mit diesem Thema, von denen nur eine Auswahl aufgeführt werden soll. Broy benutzt die Stream-Semantik, um SDL-Spezifikationen zu verifizieren [14], Damm verwendet erweiterte Statecharts [15] und Färber und Herzog annotierte SDL-Modelle und MSCs ([10], [16]). Die Erweiterungen beziehen sich auf Echtzeiteigenschaften und alle genannten arbeiten, bis auf Unterschiede bei der Darstellung von konkurrentem Verhalten, mit erweiterten, endlichen Automaten (EFSM). Mit unserer Entscheidung, SDL zu verwenden, haben wir damit Zugriff auf eine Menge von Lösungen bezüglich Realzeit. In der Regelungstechnik werden häufig Erweiterungen von Petrinetzen verwendet [17]. Ein anderer, auf temporaler Logik basierender Ansatz wurde in den Teilprojekten B4/

C1 im SFB entwickelt. Mit diesem Ansatz werden Problembeschreibungen formalisiert und überprüft [18].

In den Arbeiten von Damm und Färber werden Entwicklungsmethoden vorgestellt, die vor allem für reaktive Systeme mit harten Echtzeiteigenschaften bei sehr kleinen Reaktionszeiten entwickelt wurden. Deshalb werden als Zielplattformen FPGAs, ASICs und Signalprozessoren angenommen. Die Methoden beziehen deshalb eine High-Level-Synthese zur Generierung von Schaltungsstrukturen mit ein (klassisches HW/SW-Codesign). Unsere Zielplattformen sind wegen der geringeren Anforderungen an Reaktionszeiten in der Gebäudeautomation eher vernetzte Prozessorknoten, die andere Methoden verlangen.

Eine allgemeinere Entwicklungsmethode wird in [19] für UML vorgestellt. Da diese Methode auch einen Verfeinerungsansatz propagiert, hat sie große Ähnlichkeiten mit unserer und wir konnten viele Definitionen daraus entnehmen. Allerdings konnten wir im Detaillierungsgrad der Methode weit über [19] hinausgehen, da wir domänenspezifische Lösungen entwickeln. Außerdem ergibt sich durch die Verwendung von SDL gegenüber UML der Vorteil der frühen Ausführbarkeit.

Über die Messung der Effizienz der Methoden im Bereich reaktiver Systeme ist in der Literatur fast nichts zu finden. Es wird sich zeigen, inwieweit die internationale Fallstudie [20] mit einem von uns für die Fallstudie CS3/1 [21] entwickelten Benchmark Aufschluß bringt. Etwas mehr findet man zu Metriken und Erfahrungen beim Reuse, allerdings für allgemeine Anwendungen. Eine Übersicht in [22] zeigt, daß man hier noch sehr am Anfang steht. Allerdings haben Auswertungen der CS3/2-Ergebnisse gezeigt, daß solche Metriken im Prinzip vernünftige Ergebnisse zeigen [23].

Literatur

- [1] Mark Grand. *Patterns in Java*. Wiley. 1998
- [2] Bruno R. Preiss. *Data structures and algorithms: with object-oriented design patterns in C++*. New York: Wiley. 1999. 18.3
- [3] John Vlissides. *Pattern Hatching: Design Patterns Applied*. Addison-Wesley (Software Patterns Series). 1998
- [4] B. Geppert, R. Gotzhein, F. Rößler. *Configuring Communication Protocols Using SDL Patterns* in A. Cavalli, A. Sarma (Eds.). *SDL '97—Time for Testing*. Proceedings of the 8th SDL-Forum. Evry, France. Elsevier Science Publishers. September 1997
- [5] R.L. Feldmann, B. Geppert, F. Rößler. *Towards an Experimental Evaluation of SDL-Pattern based Protocol Design*. Technical Report No. SFB 501-04/98
- [6] *Dymola: Dynamic Modeling Laboratory*. Dynasim AB. Schweden (<http://www.dynasim.se/>)

- [7] A. Mahdavi, P. Mathew, R. Brahme, et al. *The SEMPER Project: A Multi-domain Design and Simulation Environment*. Interim quarterly report to the National Institute of Standards and Technology (12/'98-1/'99-2/'99). Built Environment Research Laboratory. School of Architecture. Carnegie Mellon University. Pittsburgh, PA. 1999
- [8] V. Pal, A. Mahdavi. *A Comprehensive Approach to Modeling and Evaluating the Visual Environment in Buildings*. School of Architecture. Carnegie Mellon University. Pittsburgh, PA. 1999
- [9] A. Metzger. *An Interlink of Building Control System Prototypes and the Lighting Simulation Lumina*. Report. Fachbereich Informatik. Universität Kaiserslautern. August 1999
(<http://www.wagz.informatik.uni-kl.de/staff/metzger/doc/interlink.ps>)
- [10] S. Petters, A. Muth, T. Kolloch, et al. *The REAR Framework for Emulation and Analysis of Embedded Hard Real-Time Systems* in Proceedings of the 10th IEEE International Workshop on Rapid System Prototyping RSP '99. Juni. 1999
- [11] J. M. Thompson, M. P.E. Heimdahl, S. P. Miller. *Specification-Based Prototyping for Embedded Systems* in SEC/FSE '99. Lecture Notes in Computer Science 1687. Heidelberg: Springer-Verlag. 1999. S. 163–179.
- [12] R. Wieringa. *A Survey of Structured and Object-Oriented Software Specification Methods and Techniques* in ACM Computing Surveys. 30(4). Dec. 98. S. 460–527
- [13] B. Selic. *Turning Clockwise: Using UML in the Real-Time Domain* in Communications of the ACM. 42(10). Oct. 99. S. 46–54
- [14] U. Hinkel. *Verification of SDL Specifications on the Basis of Stream Semantics* in Y. Lahav, A. Wolisz, J. Fischer, E. Holz (Eds.). Proceedings of the 1st Workshop of the SDL Forum Society on SDL and MSC. Humboldt-Universität zu Berlin. 1998
- [15] W. Damm, B. Josko, H. Hungar, and A. Pnueli. *A compositional real-time semantics of STATEMATE designs* in W.-P. de Roever, H. Langmaack, A. Pnueli (Eds.). Proceedings COMPOS '97. Lecture Notes in Computer Science 1536. Springer-Verlag. 1998. S. 186–238
- [16] F. Slomka. *HW/SW-Systementwurf auf der Basis von SDL* und PMSC*. Bericht IMMD7-8/97. Erlangen. 1997
- [17] E. Schnieder (Ed.). *Petrinetze in der Automatisierungstechnik*. München: Oldenbourg Verlag. 1992

- [18] C. Peper, R. Gotzhein, M. Kronenburg. *A Generic Approach to the Formal Specification of Requirements*. Proceedings of the 1st IEEE International Conference on Formal Engineering Methods (ICFEM'97), Hiroshima, Japan, November 1997
- [19] I. Jacobson, G. Booch, J. Rumbaugh. *The Unified Software Development Process*. Addison-Wesley. 1999
- [20] *Special Issue on Requirements Engineering: The Light Control Case Study*. Journal of Universal Computer Science. to be published
- [21] R.L. Feldmann, J. Münch, S. Queins, S. Vorwieger, G. Zimmermann. *Baselining a Domain-Specific Software Development Process*. Technical Report No. SFB 501-02/99
- [22] W. Frakes, C. Terry. *Software Reuse: Metrics and Models* in ACM Computing Surveys. 28(2). June 96. S. 415–435
- [23] S. Queins, G. Zimmermann. *A First Iteration of a Reuse-Driven, Domain-Specific System Requirements Analysis Process*. Technical Report No. SFB 501-13/99

3.4 Eigene Vorarbeiten

Über die eigenen Vorarbeiten wird im Ergebnisbericht ausführlich berichtet. Hier soll nur auf das Gebiet Entwurfssystematik eingegangen werden, da es ein neuer Aspekt ist.

Entwurfstheorie ist ein Zweig der Wissenschaft, für den sich vor allem auch die Künstliche Intelligenz interessiert hat. Interessanterweise wird sie in den Ingenieurdisziplinen kaum gepflegt. Unserer Arbeitsgruppe hat sich im Rahmen VLSI-Entwurf intensiv mit Entwurfstheorie beschäftigt, vor allem angeregt durch Arbeiten von Allen [24]. VLSI-Systeme sind äußerst komplex und werden hierarchisch mit vielen Alternativen und Versionen entworfen. Darin ähneln sie großen Softwaresystemen. Der Entwurf verläuft in mehreren Phasen, die auch iterativ durchlaufen werden. Wir haben dazu zunächst ein Produktmodell aufgestellt und als Datenbank implementiert, um den Prozeß zu unterstützen [25]. Dabei stellte es sich in Experimenten schnell heraus, dass ein starrer Durchlauf der Phasen nicht zum Erfolg führt. Vielmehr brauchten wir ein flexibles Verfeinerungsverfahren, um zu guten Entwürfen zu kommen. Dazu war eine Entwurfstheorie notwendig, um bei den verschiedenen Phasen, Ebenen, Komponenten und Versionen zielgerichtet vorgehen zu können. Wir haben zunächst die Theorie von Allen für unseren Prozeß erweitert [25], Reuse mit dem Problem des Selektierens geeigneter Objekte eingeführt [26] und dann alles in einer umfassenden Entwurfstheorie für VLSI zusammengebracht [27].

Die grundlegenden Ideen der schrittweisen Verfeinerung haben sich auch in der Systemanalyse für reaktive Systeme gegenüber einem starren Phasenkonzept bewährt, wie die Fallstudien CS3/1 [18] und CS3/2 [18] im Vergleich gezeigt haben. Da bei reaktiven Systemen Hard- und Software zusammen entwickelt wird, ist der Prozeß ein HW/SW-Codesign-Prozeß und es lassen sich Teile der Entwurfstheorie für Hardware auch hier verwenden. Allerdings unterscheiden sich Artefakte und Operationen und die Entwurfstheorie muß neu durchdacht werden. Wir haben dazu Vorüberlegungen angestellt, vor allem über die kleinsten Modellierungseinheiten und über Elementaroperationen, um diese Einheiten zu modifizieren. Dabei haben wir auf die Erfahrungen bei den bisherigen Entwürfen und auf die gesammelten Artefakte und Messungen zurückgegriffen. Wir sprechen hier zunächst von einer Entwurfs-Systematik, da noch nicht klar ist, ob eine Theorie daraus entsteht.

Literatur

- [24] J. Allen. *Performance-Directed Synthesis of VLSI Systems* in Proceedings of the IEEE. 1990. S. 336–355
- [25] E. Siepmann. *Entwurfstheorie und Entwurfsdatenmodellierung für CAD-Frameworks*. Dissertation. Fachbereich Informatik. Universität Kaiserslautern. 1991
- [26] J. Altmeyer. *A Refinement Design Model and its Application to Data Modeling and Retrieval* in Fortschritt-Berichte VDI. Reihe 20, Nr. 252. Düsseldorf: VDI Verlag GmbH. 1997
- [27] B. Schürmann. *ECAD-Entwurfsmanagement*. Habilitationsschrift. Fachbereich Informatik. Universität Kaiserslautern. 1997

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

Das Teilprojekt D1 hat zwei Aufgaben. Es stellt erstens in allen Phasen des Entwicklungsprozesses eine kontrollierbare, skalierbare *Testumgebung* für Experimente der Anwendungsdomäne Gebäudeautomation zur Verfügung. Zweitens sucht D1 domänenspezifische Lösungen für die *Requirements-Engineering*-Phase der Entwicklung großer, verteilter, reaktiver Systeme mit dem Ziel ausführbarer System-Requirements. Aus beiden Aufgaben resultieren Entwicklungs- und Forschungsziele. Bevor diese genauer beschrieben werden, wird gezeigt, dass die Anwendungsdomäne sehr vielfältige funktionale und nichtfunktionale Anforderungen enthält, die mit bekannten Methoden, welche in kommerzielle Entwicklungsumgebungen für Steuerungen eingeflossen sind, nicht alle erfüllt werden können. Daher stellt die Gebäudeautomation eine interessante Herausforderung für die Entwicklung großer Systeme dar.

Anwendungsdomäne: Die Anforderungen an Gebäudeautomationssysteme können weit über die Regelung und Steuerung einzelner Systeme wie Heizung, Klimatisie-

rung, Beleuchtung und Kommunikation hinausgehen. Neben einer Reduktion der Kosten kann vor allem auch die Erhöhung des Komforts und die Steigerung der Produktivität der Benutzer gefordert werden. Zusätzlich muss die Sicherheit von Personen und Sachen gewährleistet werden. In Privathaushalten vergrößern neue Unterhaltungs- und Haushaltsgeräte die Komplexität eines solchen Systems. Es gibt auch einen Trend, immer mehr Arbeit vom Büro in die Privatwohnung zu verlagern. Die Unterstützung hilfebedürftiger Personen ist eine weitere, besondere Herausforderung. Alles zusammen erfordert enge Kopplung der Steuerung sehr verschiedener Effekte und die Einbeziehung der Personen und der Gebäudeumgebung in die Steuerungsstrategien. Diese Integration steht der in der Steuerungstechnik üblichen Entkopplung und Partitionierung in kleinere Teilsysteme entgegen und erfordert neue, mächtige Methoden und Werkzeuge für die Entwicklung verteilter, reaktiver und sicherer Hardware-Software-Systeme. Gebäudeautomation ist damit ein sehr flexibles, variantenreiches Anwendungsgebiet mit sehr verschiedenen funktionalen und nichtfunktionalen Anforderungen.

Gebäudeautomationssysteme sind, außer in wenigen Modellhäusern, längst nicht auf dem gewünschten Integrationsniveau, obwohl das von den existierenden Kommunikationsstrukturen her möglich wäre. Das liegt vor allem an den hohen Systementwicklungskosten, die nicht auf große Stückzahlen verteilt werden können. Die verwendeten datenflußorientierten Komponentensysteme (z.B. Metasys) können nur für eingeschränkte Bereiche effizient Software generieren, für andere Teile und das übergeordnete System gibt es keine fertige Unterstützung. Für diese Bereiche werden Modellierungstechniken eingesetzt, die auf Zustandsgraphen (SDL, Statecharts) oder Petrinetzen basieren (siehe auch 3.3).

Modellierungsnotationen und Werkzeuge allein reichen nicht aus, um die Entwicklungskosten drastisch zu senken. Sie stellen nur Teile einer Entwicklungsmethode dar. In allen Teilprojekten des SFB wird an einer Methode für den gesamten Entwicklungsprozess gearbeitet. Änderungen, die zu einer Verbesserung der Methode führen sollen, müssen bewertet werden. Das ist bei reaktiven Systemen meistens nur dadurch realisierbar, dass Experimente in einer möglichst realen Umgebung durchgeführt werden, die zeigen, dass das Richtige (*Validierung*) richtig (*Verifikation*) entwickelt wurde. Daraus ergibt sich unsere Vision für die Testumgebung:

Vision für die Testumgebung: *Es soll möglich sein, entwurfsbegleitend alle ausführbaren Modelle, Prototypen und Zielsystemkomponenten in beliebiger Kombination in realer oder simulierter Umgebung auf Test- oder Zielplattformen mit vertretbarem Aufwand auf funktionale und nichtfunktionale Eigenschaften zu testen und Messungen durchzuführen und auszuwerten.*

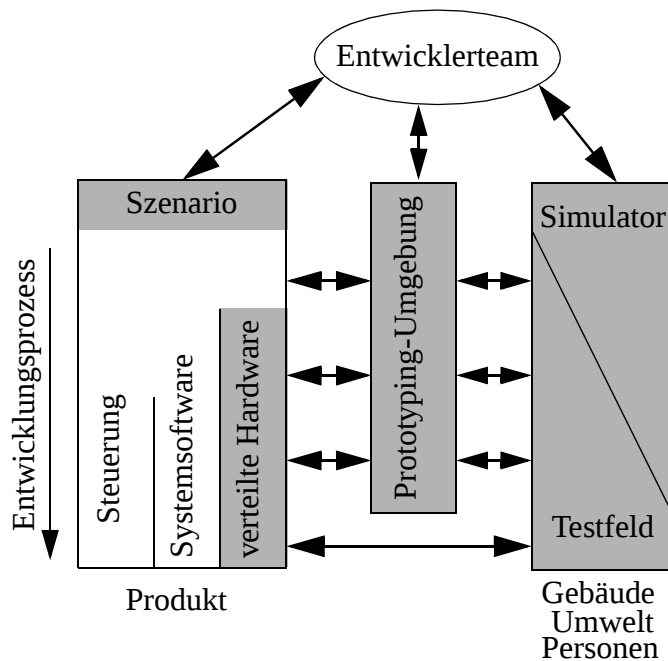


Abbildung 1: Testumgebung

Teilprojekt D1 hat als Infrastrukturprojekt die Aufgabe, *Szenarien* (Problembeschreibungen) für den Nachweis der Funktion neuer Prozess-Lösungen und der erreichten Verbesserungen zu entwickeln und eine *Testumgebung* für das Testen aller während der Entwicklung des Produktes entstandenen ausführbaren Artefakte bereitzustellen. Diese Aufgabe zerfällt grob in drei Teilaufgaben: *Testfeld*, *Simulation* und *Prototyping*, für die wir die Arbeitsschwerpunkte **A**, **B** und **C** planen. In Abb. 1 ist der Zusammenhang zwischen

den Teilaufgaben dargestellt. Die Aufgaben von D1 sind grau hinterlegt.

Das *Szenario* ist Ausgangspunkt des Systementwicklungsprozesses, der getestet werden soll. Geeignete Szenarien hängen von den Zielen der Experimente ab, die von Teilprojekten oder Teams geplant und durchgeführt werden. Im Ergebnisbericht sind eine Reihe von Experimenten aufgeführt, welche die in Abb. 1 gezeigte Infrastruktur benutzt haben. Sobald ein ausführbares Modell vorliegt, bei unserer Analysephase bereits während des Requirements Engineering, wird es für das Prototyping so erweitert, dass es über die *Prototyping-Umgebung* mit dem gesteuerten System reaktiv in Realzeit kommunizieren kann. Das gesteuerte System ist entweder ein *Simulator*, ein *Testfeld* oder die *Kombination* aus beiden. In der Analysephase werden vor allem die funktionalen Eigenschaften der *System Requirements* überprüft.

In der folgenden Design-Phase findet eine Verfeinerung des Systems in Hard- und Software und bei letzterer in Steuerungs- und Systemkomponenten (Betriebs- und Kommunikationssystem) statt. Optimierungsziele und die nichtfunktionalen Eigenschaften des Systems müssen erreicht und überprüft werden. Das ist teilweise nur auf realen, verteilten Prozessoren und Kommunikationskanälen möglich, die deshalb Teil des Testfeldes sind. An die Stelle einer reinen Simulation tritt immer mehr die Kombination mit dem Testfeld, da nur so Realzeiteigenschaften realistisch getestet werden können. Die Aufgabe des Prototyping wird damit während der Design-Phase verkompliziert. Das Steuerungssystem besteht jetzt aus einem Software-Prototyp und den zum Teil schon realen Systemsoftware-Komponenten, die auf einer Mischung aus echten Zielsystem-Komponenten und leistungsfähigen Rechnern in einer Umgebung ablaufen, die zum Teil aus Simulation und Testfeld besteht.

Aus diesen Testszenarien ergeben sich die Anforderungen und Ziele für die drei ersten Schwerpunkte.

Schwerpunkt A: Testfeld

Ziel: *Einen Teil eines Gebäudes mit Sensorik und Aktuatorik so ausstatten, dass mehrere konkurrierende physikalische Effekte in Reaktion auf Personen und die Umgebung gesteuert werden können. Die Sensoren und Aktuatoren sind mit verschiedenen Prozessorknoten über Feldbusse und LANs konfigurierbar so zu vernetzen, dass die genannten Testszenarien skalierbar ausgeführt werden können. Der Gebäudeteil soll als „living laboratory“ normal genutzt werden können. Notlaufeigenschaften für den Minimalbetrieb müssen gewährleistet sein.*

Der größte Teil des Testfeldes existiert bereits oder wird zur Zeit implementiert (siehe Ergebnisbericht). In der laufenden Periode wurde das Testfeld mit Instrumentierung für Licht, Heizung und Bewegung bereits in mehreren Experimenten zum Testen genutzt.

In der nächsten Periode soll das Testfeld so erweitert werden, dass auch Gebäudeautomationssysteme für große Gebäude erprobt werden können. Unter groß verstehen wir dabei 100 bis 1000 getrennt steuerbare architektonische Einheiten, also zum Beispiel Büros oder Arbeitsplatzbereiche. Erst bei dieser Größenordnung ergeben sich für das Automatisierungssystem Herausforderungen an Verarbeitungs- und Kommunikationsgeschwindigkeit und Fehlertoleranz, die Strukturierungsmaßnahmen erforderlich machen. Diese *Skalierung* ist mit dem Testfeld allein nicht möglich, sondern nur in Verbindung mit Simulation. Zum gleichzeitigen Betrieb von Testfeld und Simulator sind Enkopplungsbereiche notwendig, in denen Eigenschaften des Testfeldes emuliert werden. Diese Bereiche um die vorhandenen Testfeldräume sollen ausgebaut und vernetzt werden.

Hinzu kommen Anpassungs- und Installationsarbeiten zur Ergänzung des Testfeldes, wenn es zum Beispiel die von B4 und B5 entwickelten und generierten Systemkomponenten notwendig machen. Vernetzung über Funk oder Power-Line halten wir uns für eventuellen Bedarf offen.

Nachweis: Der Nachweis einer erfolgreichen Installation wird wie bisher durch Experimente des SFB in dem Testfeld gezeigt.

Schwerpunkt B: Simulatorgenerierung

Ziel: *Automatische Generierung von Simulatoren mit Hilfe von Gebäudebeschreibungen und Steuerungssystemmodellen, die alle für das Steuerungssystem notwendigen Objekte (auch Personen) und Effekte so genau und so schnell simulieren, wie es für den Nachweis von Anforderungen mit Prototyping oder für den Produkttest notwendig ist. Die Simulation muss verteilt auf der Zielplattform zusammen mit Komponenten des Testfeldes in echter und in komprimierter Zeit ablaufen können, um zum Beispiel*

reale Kommunikationsanforderungen (B4) auch durch Emulation von Gebäudeteilen in dem realen, verteilten System messbar zu machen.

Bisher wurde gezeigt, dass objektorientierte, ereignisgesteuerte Simulation für Wärmerechnungen in Gebäuden prinzipiell möglich ist und dass solche Simulatoren durch die Verwendung generischer und domänenspezifischer Design Pattern einfach hergestellt werden können. Es wurde auch gezeigt, dass die für die Modellierung der Steuerung aus der Gebäudetopologie abgeleitete Struktur weitgehend für die Simulatorstruktur wiederverwendet werden kann. Erste Messungen haben ergeben, dass aus SDL-Modellen automatisch generierter Simulator-Code auch für größere Gebäude in Realzeit ausgeführt werden kann.

In der nächsten Periode soll ein Schwerpunkt die Untersuchung der Auswirkungen verschiedener Architekturen und Architekturstile auf objektorientierte Simulatoren sein. Damit hängt auch die Verknüpfung verschiedener Modelltypen zusammen. Es geht vor allem darum, den Aufwand bei der entwurfsbegleitenden Generierung von Simulatoren im Prototyping-Prozess und die Laufzeit so gering zu halten, daß der Produktentwicklungsprozess nicht wesentlich verlangsamt wird. Diese Entwicklungseffizienz muß gezeigt werden.

Mit diesem Ziel hängt auch der zweite Schwerpunkt zusammen, für die Simulatorentwicklung einen Prozess aufzustellen, der möglichst weitgehend mit dem Produktentwicklungsprozess übereinstimmt und zu ihm synchron läuft. Unterschiede ergeben sich durch die Anwendung von Pattern, die sich wegen der eingeschränkten Domäne (Gebäudesimulation) besonders effizient einsetzen lassen. Daran soll die im Arbeitsschwerpunkt D1 entstehende Entwurfssystematik erprobt werden, da die Simulatorentwicklung letztlich ein weiterer Anwendungsfall ist. Es soll auch untersucht werden, wie sich ähnliche Pattern für die Entwicklung von Steuerungen nutzen lassen.

Drittens soll die korrekte Funktion der Simulatoren in der betrachteten Anwendungsdomäne, auch in Kombination mit dem Testfeld, gezeigt werden. Bei der Korrektheit der erzeugten Gebäudesimulatoren ist hier nicht die statische Genauigkeit ausschlaggebend, sondern eine mit der Realität übereinstimmende Reaktion auf dynamische Veränderungen von externen Parametern und auf Ereignisse. Diese können von der Steuerung und von anderen Systemen, zum Beispiel dem Testfeld oder Personen, ausgelöst werden. Neben einer im Rahmen der Anforderungen korrekten mathematischen Umsetzung des Simulationsproblems muss der Simulator mindestens in Realzeit, besser auch in komprimierter Realzeit ablaufen können, um auch über größere Simulationszeiträume schnell testen zu können.

Methode: Die Modellierungsmethode muss es einem Entwerfer ermöglichen, in kurzer Zeit einen Simulator zu erzeugen, der vorgegebenen Qualitätsmerkmalen genügt. Die Qualitätsmerkmale (z.B. Geschwindigkeit oder Simulationsgenauigkeit) leiten sich dabei aus dem jeweiligen Szenario ab. Zur Unterstützung des Entwerfers müssen eine auf die Domäne angepasste Modellierungsmethode und flexibel einsetzbare vorgefertigte Bausteine bereitgestellt werden. Die bisher eingesetzten Modellierungsle-

mente (Klassendiagramme, Pattern, Bibliotheken) eignen sich gut zum Simulatorentwurf, müssen jedoch noch weiter verfeinert und verifiziert werden. Insbesondere sollen soweit wie möglich Qualitätsmerkmale aus den Modellen ableitbar sein.

Die mathematische Umsetzung des Simulationsproblems beruht bei der von uns entwickelten objektorientierten Simulationsmethode darauf, dass auch die Ausführung in einzelnen Objekten erfolgt und nicht, wie bei anderen objektorientierten Simulatoren, eine objektorientierte Eingabe auf ein großes Gleichungssystem abgebildet wird (Beispiel Dymola [6], SEMPER [7]). Die oberen Ebenen der Simulation sind bei unserem Ansatz ereignisgesteuert, die unteren können diskret oder kontinuierlich sein. Die Korrektheit dieser Umsetzung muss insbesondere für dynamische Änderungen bei kontinuierlichen Effekten durch Vergleichsmessungen nachgewiesen werden. Zur Messung des realen Kommunikationsaufkommens muss es möglich sein den Simulator auf verschiedene Rechnerknoten zu verteilen. Dabei soll die Partitionierung entlang natürlicher Grenzen (d.h. entlang nicht oder nur schwach gekoppelter Objekte) erfolgen, um Synchronisationsprobleme gering zu halten. Da es sich bei den meisten Effekten um Ausgleichsprozesse handelt, ist eine strenge Synchronisation nicht erforderlich. Der Datenaustausch zwischen einzelnen Objekten muss innerhalb bestimmter Intervalle erfolgen, um eine korrekte Simulation zu gewährleisten. Diese Intervalle sind aber in der Regel so groß, dass einer verteilten Simulation in Realzeit nichts im Wege steht.

Nachweis: Alle geforderten Eigenschaften werden mit Experimenten überprüft und gemessen. Zum Nachweis der thermischen Simulationsgenauigkeit wird mit detaillierten Aufzeichnungen von Klimadaten in Testräumen verglichen, ob die dynamischen Eigenschaften mit einer ausreichenden zeitlichen Auflösung stimmen. Als Geschwindigkeit soll für die Simulation eines Stockwerkes mindestens Realzeit nachgewiesen werden. Eine Simulatorentwicklung soll nicht länger dauern als die des getesteten Steuerungssystems.

Schwerpunkt C: Prototyping

Wichtige Schritte im Entwurfsprozess sind die Verifikation und die Validierung. Beides wird durch Prototyping unterstützt.

Generell sollte eine Validierung so früh wie möglich erfolgen, denn werden Fehler erst später erkannt, kann dies zu teuren Rückkopplungen im Entwicklungsprozess führen. Wird das Prototyping dazu während des Requirements Engineering eingesetzt, führt dies in der Regel zu qualitativ besseren (z.B. bezüglich Vollständigkeit) System Requirements, da der Kunde frühzeitig am Entwurfsprozess beteiligt wird und dadurch Klarheit über die gewünschten Eigenschaften des Systems besteht.

Der Prototyp dient weiter als Hilfsmittel bei der Verifikation. Durch geeignete Tests lassen sich viele Eigenschaften des Systems kontrollieren, die sich ohne einen Prototyp oftmals gar nicht überprüfen ließen. Zum Beispiel kann man den Komfort, den ein

Gebäudeautomationssystem bietet, wohl erst dann beurteilen können, wenn man sich real in einem so gesteuerten Gebäude bewegt.

Ziel: Die Prototyping-Umgebung muss durch geeignete Werkzeuge und Schnittstellen zwischen Plattformen, Steuerungssystemkomponenten, Testfeld und Simulatoren sicherstellen, dass ein Entwicklungsteam neben einer effizienten Erstellung des Prototyp alle geplanten Testfälle durchspielen und Messungen mit vernünftigem Aufwand ausführen kann. Dazu muss das Gesamtsystem in Realzeit ausführbar sein, bzw. alle Komponenten synchronisiert werden. Benutzerschnittstellen müssen vorhanden sein, die Beobachtung und Eingriff erlauben. Prototyping soll möglichst früh im Prozess dadurch unterstützt werden, dass Teilsysteme durch automatisch generierte Treiber und Stubs zu ausführbaren Systemen vervollständigt werden. Prototyping soll aber auch in späteren Phasen, also entwurfsbegleitend durchgeführt werden, und dann eine Partitionierung in Teilsysteme oder die Einbindung von Hardware ermöglichen. Aussagen über konkurrentes Verhalten und nichtfunktionale Eigenschaften des Systems sollen mit Hilfe von Prototypen ermittelt werden. Dazu werden die Prototypen entsprechend instrumentiert und Modelle für die Auswertung der am Prototyp gemessenen Daten entwickelt.

[illegible]

- [3] Camacho E. F., Bordons C.: Model predictive control (Advanced textbooks in control and signal processing), Springer London Berlin Heidelberg, 1999
- [4] Dymola. Lund/Schweden: Fa. Dynasim, 1999; <http://www.dynasim.se>
- [5] Lu, Y., Arkun, Y.: Constrained robust predicitive controller for uncertain processes modeled with orthonormal series functions. In: Automatica (36) 4, 2000.
- [6] Passino K. M., Yurkovich S.: Fuzzy Control, Addison-Wesley, Menlo Park Calif., 8.Auflage 1998
- [7] Roberts, P.D.: A brief overview of model predicitive control. In: Computing & Control Engineering Journal (10) 5 , 1999.
- [8] Modelica - a unified object-oriented language for physical system modeling. The Modelica Design Group 1999; <http://www.dynasim.se/Modelica>
- [9] Otter M.: Objektorientierte Modellierung physikalischer Systeme Teil 1-5. at1, A1-A4(1999); at2, A5-A8(1999); at3, A9-A12(1999); at4, A13-A16(1999); at5, A17-A20 (1999)
- [10] Otter M., Bachmann B.: Objektorientierte Modellierung physikalischer Systeme Teil 6. at6, A21-A24(1999)
- [11] Otter M., Elmqvist H., Mattsson L. und Mattsson S.E.: Objektorientierte Modellierung physikalischer Systeme Teil 7 + 8. at9, A25-A28(1999); at11, A29-A32(1999)
- [12] Otter M., Schlegel C.: Objektorientierte Modellierung physikalischer Systeme Teil 9. at12, A33-A36(1999)

3.4 Eigene Vorarbeiten

Eine ausführliche Beschreibung findet sich im Ergebnisbericht der zurückliegenden Förderungsperiode.

Gebäudeautomationssysteme aggregieren im reaktiven Teil atomare Regelungs- und Steuerungsalgorithmen, die über Sensoren und Aktoren mit der realen Welt interagieren. Aufgabe der Algorithmen ist es, die von den Sensoren kommende Prozessinformation zu Aktorbefehlen zu verarbeiten [17]. Neben der Vielzahl zu verarbeitender Signale stellen die gegenseitigen Verkoppelungen der zu beeinflussenden Teilprozesse Herausforderungen beim Algorithmenentwurf dar. Diese Problematik berührt klassische Fragestellungen der Automatisierungs- und Regelungstechnik.

Zunächst werden in einer Analyse der Aufgabenstellung die Wirkzusammenhänge und Verkoppelungen der zu beeinflussenden Größen detektiert. Hierzu wird ein den

technischen Prozess beschreibendes mathematisches Modell erstellt. Dies ist unabdingbar für die Algorithmenentwicklung. Das Modell dient dem Zustandekommen des Algorithmus, dessen Parametrierung (Verstärkungskonstanten, Zeitkonstanten, Hysteresebreiten) sowie dem Test. Die Qualität eines Modells zeigt sich in der realitätsnahen Nachbildung aller für den Algorithmenentwurf relevanten physikalischen Eigenschaften des realen technischen Systems. Die Qualität des Algorithmus ergibt sich aus dem Grade der Erfüllung seiner Regelungsaufgabe.

Das Teilprojekt D2 setzte sich in der zurückliegenden Förderungsperiode das Ziel, die Effizienz des Entwurfes wie auch die Qualität sowohl der mathematischen Modelle wie auch der eigentlichen Regelungs- und Steuerungsalgorithmen durch Anwendung wiederverwendungsunterstützender Konzepte zu optimieren [13] - [16]. Die von uns zugrundegelegte Vorgehensweise einer modellgestützten, generischen Reglerentwicklung ist in Abbildung 1 veranschaulicht.

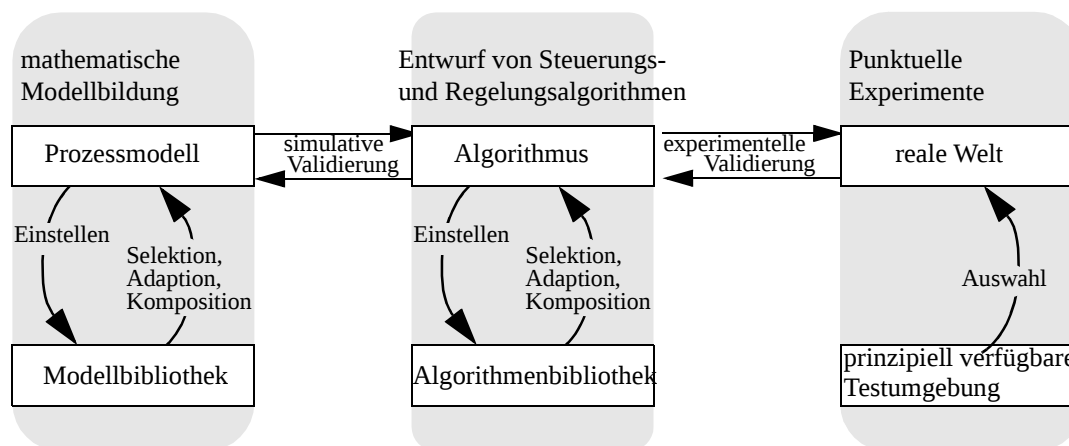


Abbildung 1: Veranschaulichung des modellbasierten Reglerentwurfes

Sowohl der Entwurf der komponentengestützten mathematischen Modelle als auch der Entwurf der Steuerungs- und Regelungsalgorithmen greifen auf Bibliotheken zurück. In den Bibliotheken liegen Komponenten, die nach Selektion und Adaption generisch zu einem Gesamtmodell bzw. Gesamtalgorithmus komponiert werden. Das Spektrum entwickelter und verfügbarer Algorithmen besteht aus parametrierbaren, komponierbaren Grundalgorithmen (PKG), die im Falle komplexer Strecken auch in mehrfach verschleifter, kaskadierter Form eingesetzt werden können, wie in Abbildung 2 exemplarisch dargestellt.

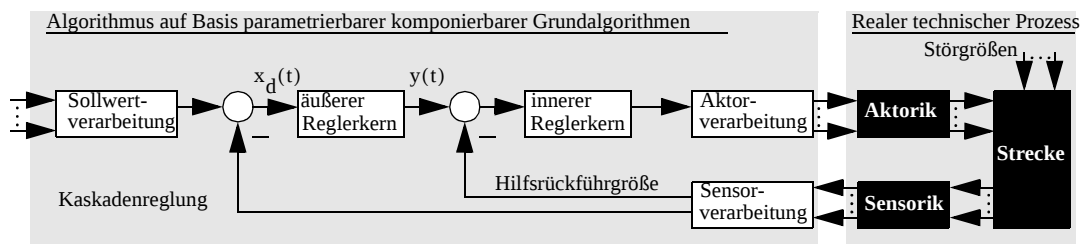


Abbildung 2: Prinzipskizze eines verschleiften Reglers aus parametrierbaren, komponierbaren Grundalgorithmen (PKG)

Exemplarische Instantisierungen wurden zur Validierung simuliert und die Ergebnisse mit validierten, kommerziellen Gebäudesimulatoren verglichen. Erste exemplarische, experimentelle Validierungen an bislang zur Verfügung stehenden Testobjekten zeigen die Verlässlichkeit der theoretischen Simulationsergebnisse.

Literaturliste

- [13] R. Merz und L. Litz: *Anwendung generischer Methoden bei der objektorientierten Modellierung großer, komplexer dynamischer Systeme*, Technical Report SFB 501- 07/99, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, 1999
- [14] J. Schäfer und L. Litz: *Objektorientierte Entwicklung von Regelungs- und Steuerungsalgorithmen für die Anwendungsdomäne Gebäudeautomation unter den Aspekten Wiederverwendbarkeit und Generik*, Technical Report SFB 501-010/99, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, 1999
- [15] J. Schäfer: *Reuse- and Object-Oriented Development of Standard Algorithms* IFAC Workshop on Digital Control: *Past, Present and Future of PID Control* 5.-7.4.2000 Vapor Universitari, Terassa, Spain 2000
- [16] Merz, R. und Litz, L.: *Objektorientierte mathematische Modellierung: Generische Methoden bei komplexen dynamischen Systemen*, Informatik Spektrum 2(2000), Springer Verlag Heidelberg, S. 90-99
- [17] L. Litz und G. Frey: *Methoden und Werkzeuge zum industriellen Steuerungsentwurf - Historie, Stand, Ausblick*, at-Automatisierungstechnik 47(1999)4, R. Oldenbourg Verlag, S. 145-156

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

a) *Teilprojekteigene Arbeitsgebiete*

Ziel des Teilprojektes ist die Entwicklung und Anwendung einer generischen, reuse-unterstützenden Methodik zum Entwurf von Regelungs- und Steuerungsalgorithmen sowie der dazu notwendigen mathematischen Modelle. Sowohl zur Erhöhung der Effektivität als auch zur Optimierung der Qualität durch Einsatz wirkungsvalidierter, getesteter Komponenten sollen bei mathematischer Modellbildung und Algorithmenentwurf gezielt reuseunterstützende Konzepte zur Anwendung kommen. Dies beginnt mit der generischen Erzeugung instantiiert mathematischer Modelle als ersten Schritt eines Entwurfprozesses. Hierauf basierend folgt der eigentliche Entwurf der Steuerungs- und Regelungsalgorithmen. Ein Schwerpunkt wird die Integration dieser Entwurfsprozesse in einen allgemeinen Softwareentwicklungsprozess sein.

Um mit modernen bautechnischen Entwicklungen Schritt zu halten, sind die Modellbibliotheken um innovative technische Komponenten zu erweitern, die überhaupt erst interessante Anwendungsfelder anspruchsvoller Regelungskonzepte ermöglichen.

Ausgangspunkt ist neben einer Gebäudebeschreibung die Problembeschreibung, zunächst aus Kundensicht. Nach formaler Spezifikation der sich hieraus ergebenden Anforderungen an das zu entwickelnde System beginnt der eigentliche Entwurfsprozess. Bei einer steuerungs- bzw. regelungstechnischen Aufgabe benötigt man zusätzlich detaillierte Angaben über den zu steuernden bzw. zu regelnden technischen Prozess, über die sogenannte Strecke. Diese umfasst auch die jeweilige Aktorik und Sensorik. Ein mathematisches Modell des technischen Prozesses leistet dies. Das Modell ist somit auch Teil der formalen Spezifikation der Regelungsaufgabe.

Der Neuentwurf eines mathematischen Modells benötigt in der Regel detailliertes Expertenwissen und fundierte Kenntnisse der Anwendungsdomäne. Die Vorgehensweise der theoretischen Analyse zur mathematischen Modellbildung ist in den Ingenieurwissenschaften prinzipiell etabliert. Die Forderung nach schnellen Entwicklungszeiten favorisiert meist top-down-Entwurfsstrategien. Ausgehend von einer hohen Abstraktionsebene wird die Beschreibung der physikalischen Realität sukzessive, soweit als notwendig detailliert. Hierbei tritt jedoch das entscheidende Defizit dieses Entwicklungskonzeptes zu Tage. Man erzeugt durch immer feingranularere Beschreibungen einer realen physikalischen Konfiguration naturgemäß Unikate. Diese sind zwar parametrierbar, beschreiben aber ausschließlich die aktuelle Konfiguration. Eine andere, ähnliche Konfiguration erfordert einen erneuten Durchlauf des kompletten Entwurfsverfahrens. Die Qualität und Effizienz beruht auf den Erfahrungen und Kenntnissen des Anwendungsexperten

Die bei der Analyse komplexer Systeme notwendige Dekomposition in Teilsysteme legt folgende Vorgehensweise nahe: Eine Bibliotheksstruktur implementierter Kom-

ponentenmodelle ermöglicht die Generierung konkreter Gesamtsysteme in einer nahezu bottom-up-ähnlichen Vorgehensweise durch Komposition bereits entwickelter, anpassbarer, parametrierbarer Bausteine. Zudem können Komponenten aus bestehenden Gesamtmodellen getauscht, ergänzt oder durch Neuentwürfe ersetzt werden. Objektorientierte Ansätze zur mathematischen Modellierung technischer Systeme [16] forcieren hierbei unter Anwendung des Klassenprinzips gezielt die Wiederverwendung. So kann durch Vererbung gemeinsamer Strukturen, Berechnungsvorschriften, Parameter und Konstanten der Implementierungsaufwand und damit die zu beherrschende Komplexität, aber auch der Aufwand zur Modifikation bestehender Modelle drastisch reduziert werden. Hier existiert jedoch eine Grenze. Eine allzu große Hierarchietiefe wirkt durch geringere Überschaubarkeit und Transparenz der Wiederverwendbarkeit eher entgegen als sie zu fördern. Aggregation als ordnendes Element erzeugt die zur Beherrschung der Komplexität notwendige Abstraktion und Hierarchie. Detaillierte Schnittstellendefinitionen und Komponentenkapselung garantieren die umgebungsunabhängige Anwendbarkeit der einzelnen Module. Nicht berechnungskausale Beschreibungstechniken unterstützen die universelle Wiederverwendbarkeit der implementierten Rechenmodelle. Entsprechende Simulatoren, welche neben der mathematischen Numerik auch geeignete Beschreibungssprachen zur Verfügung stellen, sind in den Ingenieurwissenschaften aktueller Forschungsgegenstand. In unserem Teilprojekt wurde bislang Dymola [4] der Fa. Dynasim eingesetzt. Mit Modelica [8] wird in der nächsten Periode auf einen neuen europäischen Standard übergegangen.

Die Entwicklung einer generischen Entwurfsmethodik, bei der sich das instantiierte mathematische Modell automatisch aus der Kenntnis der Regelungsaufgabe, Gebäudedaten- und struktur ableiten lässt, erhöht Qualität und Effektivität des Erstellungsprozesses drastisch. Im Idealfall sollte das Einbringen weiteren Expertenwissens an dieser Stelle überflüssig sein. Ein vorgeschlagener Lösungsansatz zur Realisierung einer solchen generischen Entwurfsmethodik mathematischer Modelle sowie der Regelungs- und Steuerungsalgorithmen erscheint vielversprechend und soll im Teilprojekt D2 entwickelt werden. Die projektierte Vorgehensweise ist in Abbildung 3 skizziert.

Im Rahmen des Querschnittprojektes QP3 („*Erarbeitung eines Domänenmodells für das Anwendungsfeld Gebäudeautomation*“) wurde ein generisches Domänenmodell entwickelt, das sowohl zum Entwurf konkreter, instantiierteter Problembeschreibungen wie auch als Referenzarchitektur für zu entwickelnde Softwaresysteme dienen kann (siehe Ergebnisbericht 1998-1999-2000). Das bestehende Dokument ist bislang anforderungsorientiert, eine weitere Detaillierung ist beabsichtigt.

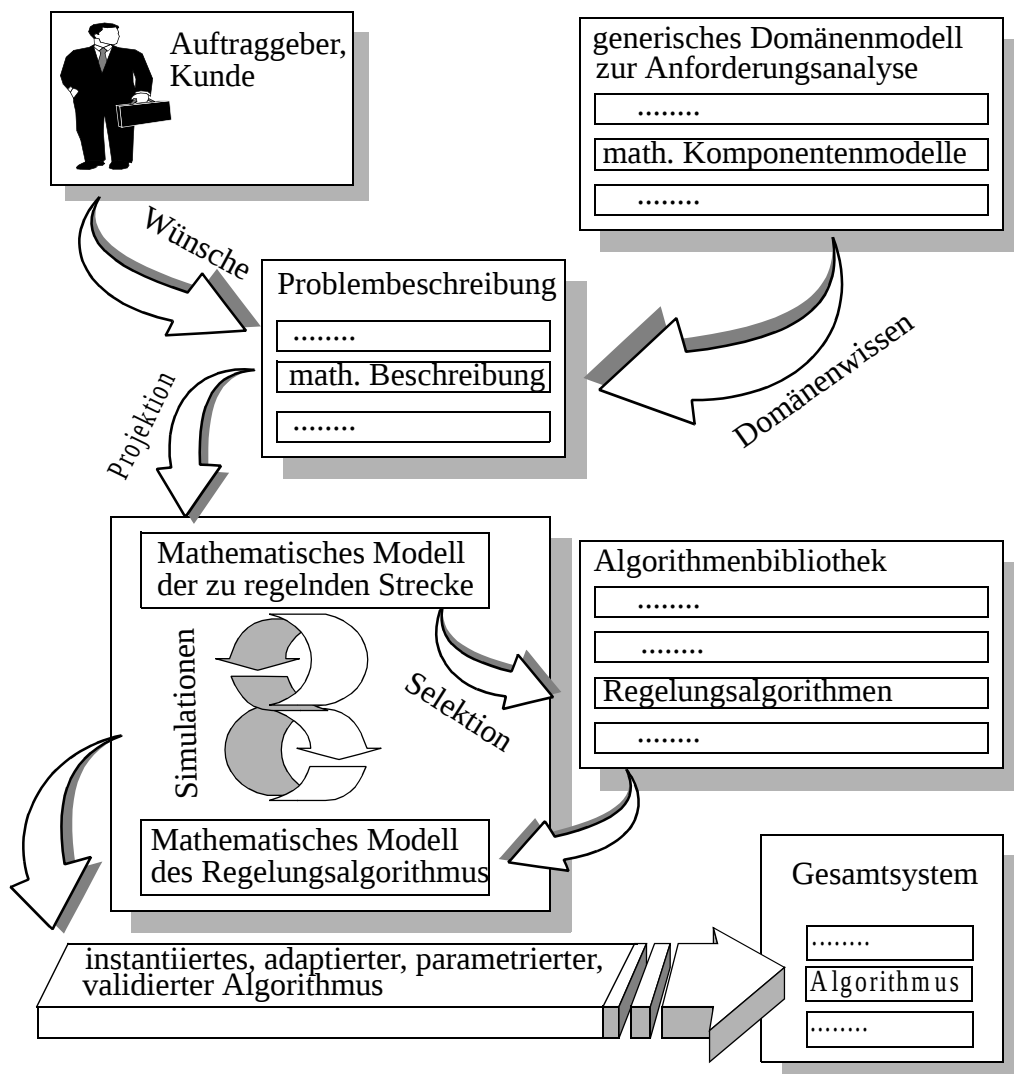


Abbildung 3: Vorgehensweise zur generische Algorithmenentwicklung

Zunächst ist eine Integration der mathematischen Beschreibungsformen des Streckenmodells in ein derartiges Dokument durchzuführen. Dem Kunden oder Softwareentwickler können diese Beschreibungsformen verborgen bleiben. Nach Instantisierung der konkreten Problembeschreibung könnte dann durch geeignete Projektion direkt das mathematische Gesamtmodell als Teil der Spezifikation der Regelungsaufgabe erhalten werden. Dieses mathematische Modell soll über Selektionsmerkmale verfügen, die es erlauben aus einer Algorithmenbibliothek mathematische Beschreibungen geeigneter Steuerungs- und Regelungsalgorithmen zu extrahieren. So ist es möglich, direkt aus der Problembeschreibung auf generische Art und Weise Simulationsmodelle bestehend aus Strecken- und Algorithmenmodellen zu erzeugen.

Simulationen ermöglichen dann die exakte Adaption und Parametrierung. Unter Umständen kann der Prozess der Selektion auch iterativ durchgeführt werden, bis der optimale Algorithmus gefunden ist. Ein wesentlicher Punkt ist die ausschließliche Verwendung validierter Modell- und Algorithmenkomponenten sowie die simulative

Validierung exemplarischer Konfigurationen mit dem Ziel zu zuverlässigen Algorithmen zu gelangen.

Die notwendigen wissensspeichernden Grundstrukturen in Form von Bibliotheken mathematischer Beschreibungen wurden in der letzten Förderungsperiode in einer ersten Version realisiert und stehen jetzt zur Verfügung. Ihre Integration in ein allgemeines Domänenmodell, das letztendlich das Wissen über die komplette Anwendungsdomäne speichert, ist vorzunehmen. Es sind verschiedene Sichten auf dieses Dokument zuzulassen, die je nach Erfordernissen die relevanten Sachverhalte herausprojizieren. Bei Bearbeitung einer regelungstechnischen Aufgabe sollen dann allein die regelungstechnisch relevanten Sachverhalte in Erscheinung treten. Es wird sich zeigen, ob dies ausreicht, bis zu einer automatischen Modellgenerierung, Reglerkomposition, -adaption und -parametrierung zu gelangen.

Neben diesen mehr methodenorientierten Zielen, die prinzipiell auch mit dem Konzept parametrierbarer komponierbarer Grundalgorithmen realisierbar wären, hat sich das Teilprojekt die Aufgabe gesetzt, die bisherigen Ergebnisse im applikativen Bereich zu vertiefen und neue Wege aufzuzeigen. So soll insbesondere der Einsatz in dieser Anwendungsdomäne bisher wenig verbreiteter höherer Regelungsalgorithmen untersucht werden. Es soll mittels detaillierter Simulationen zunächst überprüft werden, welche Verbesserungen ein höherer Regelungsalgorithmus gegenüber parametrierbaren, komponierbarer Grundalgorithmen bringt, und ob sich ein Einsatz lohnt. Die Simulationen sind durch exemplarische messtechnische Validierungen an realen Testobjekten zu validieren. Eine Erweiterung der Algorithmenbibliothek wird dann nur mit bewährten Algorithmen vorgenommen werden.

Diese Untersuchungen werden auch Aussagen zur Thematik der Selektion geeigneter Regelungskonzepte liefern. Es sind charakterisierende Attribute zu entwickeln, die es erlauben aufgrund der Aufgabenstellung und Gebäudebeschreibung geeignete Algorithmen zu selektieren. Begleitend sollen speziell für die Thematik des Teilprojektes zugeschnittene Metriken zur quantitativen Aufwandserfassung entwickelt werden. Dies geschieht mit dem Ziel, den Erstellungsaufwand konkreter zukünftiger Lösungen zu prognostizieren.

b) Teilprojektübergreifende Unterstützung als Infrastrukturprojekt

Neben diesen teilprojektspezifischen Zielen wird D2 als Infrastrukturprojekt tätig sein und insbesondere in Querschnittsprojekten Expertenwissen zur Verfügung stellen.

Bei einem realen Entwicklungsprozess verfügen die Softwareentwickler oft nicht über das benötigte Domänenwissen, das jedoch von entscheidender Bedeutung für den gesamten Entwicklungsprozess ist. Schon bei der Erstellung der Problem- und Anforderungsbeschreibung ist ein gewisser domäneneigener Erfahrungsschatz zum frühzeitigen Vermeiden von Fehlern und Inkonsistenzen notwendig. Beim eigentlichen System- und Komponentenentwurf spielt dann das konkrete Wissen über den technischen Prozess, über Regelungs- und Steuerungsalgorithmen, über physikalische

Eigenschaften von Sensoren sowie Aktoren eine entscheidende Rolle. Gleiches gilt für die Entwicklung von Testszenarien zur Validierung sowie die Beurteilung ihrer Ergebnisse.

Dieses Wissen ist in geeigneter Art und Weise von außen in den Softwareentwicklungsprozess einzubringen. Die Qualität der erstellten Software sowie die Effizienz des Entwurfsprozesses hängen in hohem Maße davon ab, inwieweit es gelingt, das benötigte Expertenwissen an geeigneter Stelle in den Entwicklungsprozess zu integrieren. Verschiedene Ansätze unterschiedlicher Effizienz sind etabliert. Diese reichen von der Integration der Domänenexperten in die Entwicklungsteams, alleinigem Einsatz vorkonfektionierter Black-boxes, über die Anwendung wissenspeichernder Datenbanken bis zur autodidaktischen Aneignung von Expertenwissen durch die Softwareentwickler selbst.

In Abbildung 4 wird die Notwendigkeit der Anbindung des Expertenwissens an einen Softwareentwicklungsprozess für reaktive Systeme veranschaulicht [16]. Exemplarisch wurde die Vorgehensweise nach dem V-Modell gewählt. Dies beinhaltet eine zunehmende Detaillierung der Dokumente bzw. Produkte ausgehend von der initialen Problembeschreibung. In allen Phasen existiert dabei ein zusätzlicher Informationsstrom von Expertenwissen, der mit zunehmender Detaillierung des Produktes quantitativ den Informationsfluss aufgrund des jeweils vorhergehenden Dokumentes sogar übertrifft.

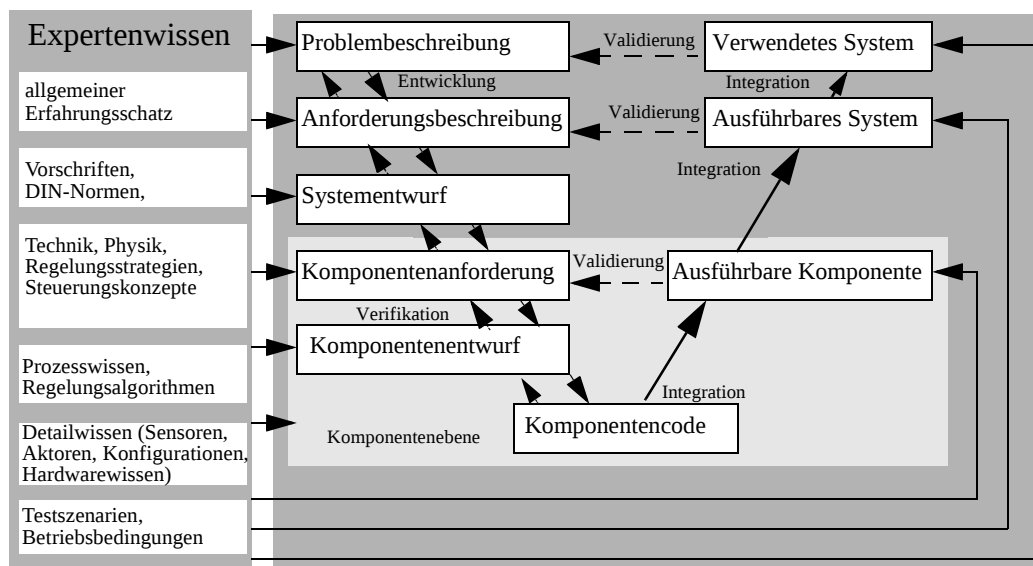


Abbildung 4: Integration von Expertenwissen in allen Phasen des Softwareentwicklungsprozesses, hier exemplarisch dargestellt am Beispiel des V-Modells.

Neben der Teilnahme an Querschnittsprojekten wird das Teilprojekt D2 allgemein nutzbares Domänenwissen wiederverwendbar in die Erfahrungsdatenbank des SFBs einbringen. Dies umfasst Regeln, Vorschriften, DIN-Normen, physikalisch-technisches Wissen, übergeordnete Regelungs- und Steuerungskonzepte, -strategien oder -ziele. Hierzu gehört auch der Ausbau der teilprojekteigenen Modell- und Algorithmenbibliotheken wie einer Bibliothek prototypischer Simulationsmodelle, die auch in die Erfahrungsdatenbank des SFBs einfließen werden. Geeignete kommentierte Darstellungen werden es erlauben die wesentlichen Zusammenhänge, Gleichungen und Informationen zu erkennen und auch gegebenenfalls Algorithmen für konkrete Anwendungen zu extrahieren.

3.5.2 Arbeitsprogramm

Der Folgeantrag gliedert sich vom logischen und zeitlichen Verlauf in die Teile (AP-A) bis (AP-D). Schwerpunkt des Teils (AP-A) ist die Ausweitung des Problemspektrums im Bereich mathematischer Modellbildung sowie die Anbindung der mathematischen Beschreibungsformen an das Domänenmodell. Der Arbeitsschwerpunkt (AP-B) befasst sich mit der generischen Erzeugung höherer Regelungskonzepte sowie mit einer vergleichenden Untersuchung. Im Arbeitsschwerpunkt (AP-C) werden Ähnlichkeitsmaße zur Selektion höherer Regelungsalgorithmen entwickelt. Im Arbeitsschwerpunkt (AP-D) sollen Metriken für die Effektivität von Reuse oder Neuentwurf anhand der konkreten Anwendungen im Teilprojekt entwickelt werden.

Arbeitsschwerpunkt A: Ausweitung des Problemspektrums im Bereich mathematischer Modellbildung

Die Ausweitung des Problemspektrums ist Voraussetzung um auf breiter Basis die Untersuchungen im Arbeitsschwerpunkt (AP-B) zu ermöglichen. Der Arbeitsschwerpunkt (AP-A) untergliedert sich in die Teile (AP-A.1) bis (AP-A.4).

Zunächst muss in Arbeitspaket (AP-A1) eine Konversion aller vorhandener Modelle von der Simulationssprache Dymola in die neue Norm Modelica erfolgen. Die ebenfalls in Dymola spezifizierten Regelungsalgorithmen sind kompatibel zu halten und daher ebenfalls zu übersetzen.

Arbeitspaket A1: Der Weg von Dymola zu Modelica

Im Bereich mathematischer Modellbildung und Simulation existiert bislang eine große Vielfalt von Beschreibungssprachen. Ihre Verbreitung ist auf eng begrenzte Anwendungsdomänen beschränkt. Die Sprachen sind spezifisch für einzelne Simulatoren. Reuse erfolgt - trotz der prinzipiellen Möglichkeit durch die Existenz von Komponentenbibliotheken - allenfalls lokal, und ein Austausch zwischen verschiedenen Entwicklungsgruppen kommt praktisch nicht vor. In jüngster Zeit wird daher der Versuch unternommen, die verschiedenen Ansätze objektorientierter Modellierungssprachen zu einer einheitlichen, gemeinsamen Sprache „Modelica“

zusammenzuführen. Nur so kann der Austausch sowie die Reuse-orientierte Nutzung entwickelter Komponenten und Komponentenbibliotheken forciert werden. Die Modelica Design Group, ein technisches Komitee innerhalb EUROSIM, versucht gerade jetzt durch Modelica einen gemeinsamen Standard zu definieren. Die erste Modelica Version wurde 1997 erstellt, eine erweiterte Version 1.1 folgte Ende 1998. Dymola bietet seit Sommer 1999 eine „im Prinzip automatische“ Übersetzung zu Modelica. Allerdings verfügt Modelica über einen größeren Sprachumfang. Um diesen auszuschöpfen, ist es erforderlich den primären Entwurf jeder Komponente diesbezüglich zu überdenken und gegebenenfalls konzeptionell zu überarbeiten.

Verfügbare Modelica Bibliotheken im Bereich thermischen Gebäudeverhaltens existieren bislang nicht. Die Komponenten sind daher selbst zu erstellen bzw. nach Modelica zu konvertieren. Auch die Regelungs- und Steuerungsalgorithmen sind zur Wahrung der Kompatibilität nach Modelica zu übersetzen. Daher findet sich AP-A1 bei allen Mitarbeitern als erste Tätigkeit im Zeitplan (3.5.3).

Personalbedarf: 7 PM

Arbeitspaket A2: Weiterentwicklung der Bibliothek mathematischer Modelle

Es zeigte sich, dass die objektorientierte mathematische Modellierung als neue Methodik einen komplett neuen Ansatz der wärmetechnischen Beschreibung erfordert. Zur Etablierung der Methodik lag der Schwerpunkt der Untersuchungen in der letzten Förderperiode bei Modellen häufig genutzter Standardkomponenten. Modelle diverser thermischer Komponenten sowie der Fluidodynamik wurden bislang implementiert und stehen jetzt zur Verfügung. Nach Erarbeitung dieser Methodik ist folgerichtige Konsequenz ihre Anwendung auf Systeme höherer Komplexität, die als Ausgangspunkt für Arbeitspaket (AP-B) benötigt werden. So ist eine Ausdehnung der Bibliothek auf innovative, technische Neuentwicklungen (Erdreichwärmespeicher, solare Fassaden, Latentwärmespeicher) projektiert. Sowohl zur experimentellen Unterstützung als auch zur Modellierung ist eine Zusammenarbeit mit dem Fraunhoferinstitut für solare Energiesysteme in Freiburg geplant, zu dem bereits diesbezügliche Kontakte bestehen. Eine solartechnische Testanlage sowie experimentelle Erdreichwärmetauscher des Lehrstuhls für Bauphysik (Professor Heinrich) an der Universität Kaiserslautern sollen ebenfalls zur Validierung der mathematischen Modelle herangezogen werden.

Personalbedarf: 38 PM

Arbeitspaket A3: Erweiterungen zur hybriden Modellbildung

Kennzeichnend für die Anwendungsdomäne des SFB 501 ist neben der Größe und Komplexität der zu entwickelnden Systeme die Interaktion mit realen, technisch physikalischen Prozessen. Sowohl der dynamische Charakter, wie die allgegenwärtigen Verkopplungen, stellen Herausforderungen beim Entwurf geeigneter Regelungs- und Steuerungsalgorithmen dar. Essentiell ist daher die Analyse des dynamischen Verhaltens der Komponenten und des gesamten zu beeinflussenden physikalischen Systems

durch eine detaillierte Modellbildung und Simulation. In der letzten Förderperiode wurden ausschließlich kontinuierliche Prozesse untersucht. Es zeigte sich jedoch, dass gerade durch das für die Anwendungsdomäne typische Wechselspiel diskreter Schaltvorgänge der Aktorik, kontinuierlicher dynamischer Prozesse und der Vielzahl auftretender Verkopplungen eine hybride, d.h. diskret-kontinuierliche Modellbildung erforderlich ist. Daher ist projektiert, insbesondere deren Interaktion zu untersuchen und die Modellbibliotheken durch entsprechende Komponentenmodelle aufzurüsten. Von besonderem Interesse ist hierbei die Auswahl einer geeigneten, formalen Spezifikation der diskreten Anteile der Steuerungsalgorithmen. Zur Simulation müssen beide Anteile, der diskrete Anteil, sowie die stärker mathematikorientierten, kontinuierlichen Regelungsalgorithmen gemeinsam mit dem mathematischen Modell der zu regelnden Strecke laufen.

Personalbedarf: 8 PM

Arbeitspaket A4: Die Integration der mathematischen Beschreibungen in das Domänenmodell

Ausgestattet mit diesen Erweiterungen ist eine Integration der mathematischen Beschreibungen des dynamischen Verhaltens der Streckenkomponenten in ein generisches Anforderungsdokument, wie es beispielsweise in QP3 („*Erarbeitung eines Domänenmodells für das Anwendungsfeld Gebäudeautomation*“) erstellt wurde, erforderlich. Bei Instantiierung des Domänenmodells in einer konkreten Anwendung sollte es so möglich sein, zugleich ein mathematisches Simulationsmodell zu erzeugen. Dieses dient dann, zusammen mit den formal spezifizierten Anforderungen dem Anwendungsexperten zum Regelungsentwurf. Das Simulationsmodell kann zudem mit Parametern ausgestattet sein, welche direkt die Anwendung geeigneter Algorithmen aus der Algorithmenbibliothek vorschlagen. Daher greift Arbeitspaket AP-A4 zusammen mit Arbeitsschwerpunkt AP-C und baut auf dessen Ergebnissen auf.

Wesentlich für die weitere Vorgehensweise ist, dass sowohl die mathematischen Modelle als auch die Algorithmen in einer simulationsfähigen und kompatiblen Form abgelegt sind, denn beide müssen zu einem Gesamtsimulationsmodell verschaltbar sein. Dieses Gesamtmodell wird in simulativen Tests genutzt um die primäre Algorithmenauswahl zu überprüfen, sowie die Adaption und Parametrierung durchzuführen. Vorausgesetzt alle abgelegten Algorithmen sind simulativ validiert, teilweise in exemplarischen Konfigurationen auch experimentell validiert, sollte die Simulation des Gesamtsystems als Wirkungsvalidierung genügen, zuverlässige, sofort einsetzbare Algorithmen zu erhalten.

Die geeignete Einbindung der mathematischen Beschreibungen unter Berücksichtigung der Schnittstellendefinitionen zu den getrennt abgelegten Algorithmenmodellen wird zentraler Forschungsgegenstand von Arbeitspaket AP-A4 sein.

Personalbedarf: 10 PM

Arbeitsschwerpunkt B: Erzeugung höherer Regelungsalgorithmen und Abschätzung des Nutzens gegenüber dem Konzept parametrierbarer, komponierbarer Grundalgorithmen (PKG)

Arbeitspaket B1: Erzeugung modellbasierter höherer Regelungsalgorithmen

In AP-B1 sollen modellbasierte höhere Regelungsalgorithmen, wie z.B. prädiktive Regler, Entkopplungsregler, Zustandsraumregler auf Basis der in Arbeitspaket AP-A2 zur Verfügung gestellten mathematischen Modelle entworfen werden. Höhere Algorithmen werden bislang allenfalls als eigenständige, monolithische Programme implementiert. Die Flexibilität solcher Lösungen ist gering. Ein Schwerpunkt der Untersuchungen wird neben der Algorithmenentwicklung sein, gezielt wiederverwendungsunterstützende Konzepte anzuwenden. Diese sollen nicht nur den Entwurfsprozess betreffen, sondern möglichst weit in die Wiederverwendung von Codefragmenten vorstoßen. Dies kann z.B. bei den prädiktiven Reglern durch geeignete Separation des Modellteils vom Optimierungsteil gelingen. Für Entkopplungs- und Zustandsraumregler sind ebenfalls geeignete Separationsmechanismen zu entwickeln.

Personalbedarf: 12 PM

Arbeitspaket B2: Entwicklung von Fuzzy-Reglern

In AP-B2 sollen Regler auf Basis von Fuzzy-Control entwickelt und implementiert werden. Wegen der gegenüber höheren Algorithmen nach AP-B1 völlig anderen mathematischen Behandlung der Fuzzy-Regler, erfordern diese eigene Untersuchungen in Bezug auf die Wiederverwendbarkeit. Während der Mechanismus der Regelabarbeitung (Teile Aggregation, Inferenz, Akkumulation und Defuzzifizierung) ohne großen Aufwand wiederverwendbar gemacht werden kann, gilt dies nicht für die applikationsspezifischen Teile Struktur der Regelbasis und Menge der Zugehörigkeitsfunktionen. Der Hauptteil der Untersuchungen in AP-B2 soll sich daher auf diese Teile erstrecken. Ziel ist es, aus dem Domänenmodell möglichst geradlinig die Grundstruktur der Regelbasis sowie die Zugehörigkeitsfunktionen für bestimmte Regelungsaufgaben zu erstellen.

Personalbedarf: 10 PM

Arbeitspaket B3: Nutzung des PKG-Konzeptes

Das AP-B3 legt die Grundlage für vergleichende Untersuchungen zwischen den höheren Regelalgorithmen und dem PKG-Konzept aus der letzten Förderperiode. Ziel ist es hier, die in AP-B1 und AP-B2 speziell für höhere Algorithmen bzw. für Fuzzy-Regler ausgewählten Applikationen so gut wie möglich mit Hilfe der PKG zu lösen. Es wird also angestrebt, mit den zunächst für Standardaufgaben entwickelten PKG möglichst dieselbe Funktionalität wie mit den Algorithmen aus AP-B1 und AP-B2 zu erzeugen.

Personalbedarf: 8 PM

Arbeitspaket B4: Vergleichenden Untersuchungen

Die PKG aus AP-B3 treten in AP-B4 in Konkurrenz zu den modellbasierten, höheren Regelungsstrategien aus AP-B1 und AP-B2. Möglichkeiten und Grenzen sowohl des PKG-Ansatzes als auch der höheren Algorithmen sollen in vergleichenden Untersuchungen erforscht und gegenübergestellt werden. Die Untersuchungen sollen nicht nur simulativ durchgeführt, sondern auch experimentell unterstützt werden. Eine kompatible und etablierte Hardware-Anbindung ist als Ergänzung zur bislang verwendeten Simulationsumgebung kommerziell verfügbar. Geeignete I/O-Boards ermöglichen es, eine reale Testumgebung messtechnisch anzubinden und aktiv zu steuern.

Personalbedarf: 30 PM

Arbeitsschwerpunkt C: Ähnlichkeitsbetrachtungen zur Selektion modellfreier und modellbasierter Regelungsalgorithmen aus der Algorithmenbibliothek

Letztendlich entspricht die Lösungsmethodik zu diesem Punkt dem fallbasierten Schließen. Eine neue regelungstechnische Aufgabenstellung erfordert zunächst das Durchsuchen einer Bibliotheksstruktur bzw. Falldatenbasis nach Lösungen ähnlicher Probleme. Es folgt die Adaption der aufgefundenen Lösung an die aktuelle Aufgabenstellung. Im Falle des Auffindens nur von Teilen oder einzelner evtl. unvollständiger Komponentenbausteine folgt zusätzlich die Ergänzung und der Neuentwurf fehlender Module. Diese sind dann zu einem Gesamtsystem zu komponieren.

Hat man so eine Lösung für das konkrete Problem gefunden, integriert man sie in die Bibliotheksstruktur bzw. Falldatenbasis zur potentiellen Wiederverwendung. Die Spezifika der aktuellen Anforderungen und Charakteristika der konkreten Strecke, die ihrer Entwicklung zugrunde lagen, sind hinzuzufügen.

Arbeitspaket C1: Selektionskriterien modellfreier Regelungsalgorithmen.

Die generische Entwicklung der modellfreien PKG Regelungsalgorithmen beruht auf der Selektion wiederverwendbarer, adaptierbarer Komponentenbausteine, die zum Gesamtalgorithmus komponiert werden. Verfügbare Information an dieser Stelle ist das mathematische Streckenmodell sowie die formal spezifizierten Kundenwünsche. Das Streckenmodell enthält insbesondere die Anzahl und die Art der verwendeten Sensoren und Aktuatoren, auf denen ein großer Teil der Selektionskriterien beruhen wird.

Personalbedarf: 3 PM

Arbeitspaket C2: Selektionskriterien höherer modellbasierter Regelungsalgorithmen.

Im Falle modellbasierter höherer Algorithmen stellt das Selektionsproblem eine deutlich größere Herausforderung dar. Hier werden die zur Selektion verwendeten, als Attribute abgespeicherten Charakteristika eine höhere Qualität der Detaillierung enthalten müssen als in AP-C1. So ist sicherlich eine detaillierte formale Beschreibung der für die ursprüngliche Auswahl des Regelungskonzeptes ausschlaggebenden Regelstreckenspezifika notwendig. Auch werden aufgrund der direkten Einbindung mathematischer Streckenmodelle in den Regelungsalgorithmus direkte Verknüpfungen zur Bibliothek mathematischer Modelle erforderlich sein.

Personalbedarf: 6 PM

Arbeitsschwerpunkt D: Untersuchungen zur Prognose der Effektivität von Komponenten-Reuse bei der Entwicklung modellfreier (PKG) und modellbasierter Regelungskonzepte

Neben der Verhaltensbewertung der implementierten Algorithmen ist ein weiteres wesentliches Kriterium der Generierungsaufwand eines konkreten Algorithmus im Anwendungsfall. Zentraler Untersuchungsgegenstand des Arbeitsschwerpunktes (AP-D1 und AP-D2) wird daher die quantitative Beurteilung der Effektivität reuseunterstützender Entwurfsmethoden sein. Es geht hierbei um den Reuse von Regelungsalgorithmen bei geänderten Aufgabenstellungen. Ziel ist es, eine Metrik (AP-D3) zur Verfügung zu stellen, die bei neuen Entwicklungsprozessen Prognosen zum notwendigen Aufwand liefert. Sie soll dann als Entscheidungshilfe dienen, ob durch Neuentwurf oder durch Reuse auf Gesamtsystemebene bzw. auf Komponentenebene in der konkreten Anwendung die höchste Effektivität zu erwarten ist.

Im Arbeitsschwerpunkt (AP-C1 und AP-C2) entwickelte Ähnlichkeitsmaße werden dazu dienen, den zu erwartenden Adaptionaufwand vorab zu schätzen. Bei allen Untersuchungen, insbesondere bei AP-B1-4 ist daher der Aufwand zur Erzeugung eines instantiierten Algorithmus zu erfassen und projektbegleitend zu registrieren. Die Erfahrungen bezüglich des quantitativen, typischen Aufwandes können dann dazu dienen, den Nutzens eines Einsatzes zu prognostizieren.

Arbeitspaket D1: Effektivität reuseunterstützender Entwurfsmethodiken bei modellfreien Regelungsalgorithmen

Die Algorithmen des PKG-Konzeptes enthalten im Gegensatz zu den höheren Algorithmen kein inneres Modell des technischen Prozesses. Schon von daher sind sie universeller wiederverwendbar. Maßgeblich für das Reuse werden vorrangig die angetroffene Sensor-/Aktuatorumgebung sowie die Aufgabenstellung sein. Zur Abschätzung der Effektivität des Reuse sollen alle qualitativen Kriterien eruiert und systematisch zusammengestellt werden.

Personalbedarf: 3 PM

Arbeitspaket D2: Effektivität reuseunterstützender Entwurfsmethodiken bei modellbasierten Regelungsalgorithmen

Die Adaption einer existierenden Reglerkomponente der höheren Algorithmen erfordert, ähnlich wie der Neuentwurf einen vorab meist unbekannten, zunächst schwer prognostizierbaren Aufwand. Der reuseorientierte Neuentwurf einer Reglerkomponente kann aufwandsintensiver sein als ein regulärer Entwurf. Erst ab einer gewissen Reusewahrscheinlichkeit zahlt sich dieser Mehraufwand in einer erhöhten Effektivität des Entwicklungsprozesses aus. Kriterien zur Abschätzung basierend auf den Erfahrungen aus *AP-B1*, *B2*, *B4* und den Selektionskriterien aus *AP-C2* werden in *AP-D2* entwickelt.

Personalbedarf: 5 PM

Arbeitspaket D3: Quantitative Metriken zur Reuse-Nutzung innerhalb des Teilprojektes

Neben diesen Aufwandsbetrachtungen sollen Aussagen zur Intensität der Reuse-Nutzung innerhalb des Teilprojektes gemacht werden. Hierzu sind gezielt Metriken zur quantitativen Fassung der Reuse-Häufigkeit für die hier betrachtete Anwendungsdomäne zu entwickeln. Messbare Charakteristika des Reusegebrauches beim Entwurf eines neuen Algorithmus könnten hierbei der quantitative Anteil des aus den Bibliotheken entnommenen Codes, die Anzahl der entnommenen Komponenten oder die Anzahl der Objekte, die von Komponenten der Bibliotheken abgeleitet wurden, darstellen. Hierdurch wird es möglich, quantitative Aussagen über die Häufigkeit der Wiederverwendung innerhalb verschiedener Entwurfsprozesse zu vergleichen und deren Effektivität zu diskutieren. Es können aber auch häufig genutzte Komponenten detektiert werden, um diese im Hinblick auf Reuseunterstützung gezielt zu optimieren.

Personalbedarf: 4 PM

3.5.3 Zeitplan

Personal	2001 1. H.J.	2001 2. H.J.	2002 1. H.J.	2002 2. H.J.	2003 1. H.J.	2003 2. H.J.
Grundausstattung						
A. Diehl, wiss. Mit.	A1,A2, A3 und A4					
				B4		
						D3
P. Denu, wiss. Mit.	B1			B4		
		C2			D2	
						D3
H. Voos, wiss. Mit.	B2			B4		
		C2			D2	
						D3
Ergänzungsausstattung						
Dr. R. Merz, wiss. Mit.	A1,A2, A3 und A4					
				B4		
						D3
NN., wiss. Mit.	B1, B2, B3			B4		
		C1			D1	
						D3
NN., stud. Hilfskraft	A1,A2, A3 und A4					
						D3
NN., stud. Hilfskraft	A1					
		B1, B2, B3 und B4				
						D3

Im Zeitplan ist ein Aufwand von wenigstens 8 Std. pro Woche und pro wissenschaftlichen Mitarbeiter der Ergänzungsausstattung für SFB-übergreifende Themenstellungen berücksichtigt, die jedes Teilprojekt zur Verfügung stellt. Diese Arbeiten werden in Teams organisiert und im Arbeitsbericht gesondert beschrieben.

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Das Teilprojekt nimmt die Rolle eines Domänenexperten wahr und wird darin von D1 (*Anwendungssystem Gebäude*) ergänzt. Insbesondere im Bereich der allgemeinen Verwaltung von Domänenwissen, sowie bei der Betreuung der realen Testumgebungen, zum Prototyping und Testen der binären Steuerungssysteme (Lichtsteuerung, Anwesenheitskontrolle,...) werden wesentliche Beiträge von D1 geleistet. D2 legt seine Schwerpunkte auf physikalisch anspruchsvolle dynamische Prozesse, deren mathematische Simulation sowie regelungstechnische Beeinflussung.

Das Teilprojekt D2 trägt insbesondere in zukünftigen Querschnittsprojekten Expertenwissen zur Steuerungs- und Regelungstechnik, Heizungs-, Klima- und Lüftungstechnik bei. Hierzu zählt auch physikalisch-technisches Wissen sowie Erfahrungen zu realen Installationen, gesetzlichen Vorschriften und Normen.

Das für die Entwicklung hier relevanter Systeme erforderliche Expertenwissen soll in die Gesamt-Erfahrungsdatenbank des SFBs eingebunden werden. Die Form der von D2 erstellten Einträge wird in Absprache mit den Teilprojekten A1 (*SE-Labor*) und A3 (*Objekt-relationale Datenbanktechnologie zur Unterstützung des Softwareentwicklungsprozesses*) entwickelt.

Das von D2 einzustellende Wissen umfasst auch die in D2 zunächst lokal erstellten Modell- und Algorithmenbibliotheken. Gerade bei der Konversion in eine geeignete Form, welche die teilprojektübergreifende Anwendbarkeit aller Ergebnisse garantiert, bietet sich eine Zusammenarbeit mit dem Teilprojekt C1 (*Formale Beschreibungstechniken*) an. Das Teilprojekt C1 wird zudem Unterstützung bei der Formalisierung sowohl der initialen Problembeschreibungen wie auch der Modell- und Algorithmenkomponenten inklusive der sie charakterisierenden Selektionsparameter leisten.

3.7 Ergänzungsausstattung für das Teilprojekt

	Bewilligung 2000		2001			2002			2003		
	Verg.-Gr.	Anz.	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM	Verg.-Gr.	Anz.	Betrag in DM
PK	BAT IIa	2	BAT IIa	2	208.800	BAT IIa	2	208.800	BAT IIa	2	208.800
			StdHkr.	2	40.800	StdHkr.	2	40.800	StdHkr.	2	40.800
	zus.:	2	zus.:	4	249.600	zus.:	4	249.600	zus.:	4	249.600
SV			Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM	Kostenkategorie oder Kennziffer		Betrag in DM
			(515)		-	(515)		-	(515)		-
			(522)		19.000	(522)		19.000	(522)		12.000
			zusammen		19.000	zusammen		19.000	zusammen		12.000
I			Mittel für Investitionen insges.			Mittel für Investitionen insges.			Mittel für Investitionen insges.		
			-			-			-		

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

3.7.1 Begründung des Personalbedarfs

	Name, akad. Grad, Dienststellung	engeres Fach des Mitarbeiters	Bezeichnung des Institutes oder der Hochschule	Anteil der aufgew. Gesamt- arbeitszeit (bera- tend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Ver- gütungs- gruppe
Grundausstattung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Litz, L.; Prof. Dr.-Ing.	Automat.	FB EIT, Uni KL	5	1.6.98	C4
	(2) Denu, P., Dipl. Ing.	Automat.	FB EIT., Uni KL	10	1.6.98	at
	(3) Diehl, A., Dipl. Ing.	Automat.	FB EIT, Uni KL	20	1.6.98	at
	(4) Voos, H., Dipl. Ing.	Automat.	FB EIT., Uni KL	10	1.6.98	at
Ergänzungsausstattung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Merz, R., Dr.	Automat.	FB EIT, Uni KL	38.5	1.6.98	IIa
	(2) N.N.	Automat.	FB EIT, Uni KL	38.5		IIa
	(3) N.N., StdHkr.	Automat.	FB EIT, Uni KL	19		
	(4) N.N., StdHkr.	Automat.	FB EIT, Uni KL	19		
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Eine explizite Aufgabenzuordnung der Mitarbeiter der Grundausrüstung ist dem "Zeitplan" zu entnehmen.

Herr Diehl befasst sich im Rahmen seiner Dissertation mit der Automatisierung kältetechnischer Anlagen. Seine Aufgaben im SFB 501 untergliedern sich in zwei Bereiche. Im Bereich der Modellbildung (AP-A1-3) wird er bei der mathematischen Modellierung der klimatechnischen Komponenten unterstützen. Im Bereich Reglerentwicklung (AP-B4) wird er die experimentellen, vergleichenden Untersuchungen und Validierungen mit Expertenwissen zur Sensor- und Aktoranbindung versorgen.

Die Herren Denu und Voos werden im Bereich Reglerentwicklung (AP-B1 bzw. AP-B2) die Entwicklung der höheren Regelungsalgorithmen vorantreiben. Herr Denu befasst sich im Rahmen seiner Dissertation mit prädiktiven Reglern, Herr Voos mit Markt-basierten Kontrollstrategien, sowie Fuzzy Control. Daher wird Herr Denu AP-B1, Herr Voos AP-B2 bearbeiten. In AP-B4 folgen die vergleichenden Untersuchungen simulativ sowie experimentell. Selektionsmaße zu ihren Algorithmen werden von beiden Mitarbeitern in AP-C2 entwickelt, Betrachtungen zur Effektivität schließen sich in AP-D2 an.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Eine explizite Aufgabenzuordnung der Mitarbeiter der Ergänzungsausrüstung ist dem "Zeitplan" zu entnehmen. Es findet sich dort eine detaillierte Zuordnung und Aufteilung der Aufgaben der einzelnen Arbeitspakete.

Der erste wissenschaftliche Mitarbeiter aus der Ergänzungsausrüstung befasst sich vorwiegend mit der Thematik des Algorithmenentwurfes. Seine Aufgabe ist es, zunächst beim Entwurf der höheren Regelungsalgorithmen in AP-B1 und AP-B2 die Arbeiten von Herrn Denu und Herrn Voss durch Integration generischer Konzepte zu unterstützen. Parallel dazu entwirft er in AP-B3 eigenständig modellfreie Regelungskonzepte auf Basis des innerhalb der letzten Förderungsperiode zur Verfügung gestellten PKG-Konzeptes zum Vergleich mit den höheren, modellbasierten Algorithmen aus AP-B1 und AP-B2. Er vertritt dann diese Regelungskonzepte in den vergleichenden Untersuchungen AP-B4. Selektionsmaße betreffend seine Algorithmen werden von ihm in AP-C1 entwickelt. Betrachtungen zur Effektivität schließen sich in AP-D1 an.

Dr. Merz ist als Physiker verantwortlich für den Bereich mathematische Modellbildung (AP-A1-4). Neben der Erweiterung des Spektrums mathematischer Modelle in AP-A2 und AP-A3 stellen simulative und experimentelle Validierungen einen Schwerpunkt seiner Untersuchungen dar. Dies erstreckt sich auch auf die Interaktion der mathematischen Streckenmodelle und der neuen Regelungskonzepte. Er übernimmt daher in den Arbeitspaketen AP-B4 die Koordination der vergleichenden Untersuchungen. Er wird sich in AP-A4 mit der Einbindung der regelungstechnischen Vorgehensweise in allgemeine Softwareentwurfsprozesse befassen.

Die erste studentische Hilfskraft unterstützt den Bereich Modellbildung (AP-A), die zweite studentische Hilfskraft den Bereich Reglerentwicklung (AP-B).

Alle Mitarbeiter führen in AP-D3 begleitende Messungen und Betrachtungen zur Wiederverwendungshäufigkeit bei ihren eigenen Tätigkeiten durch. Eine Analyse erfolgt zu Ende des Förderungszeitraumes.

3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

	2001	2002	2003
Für sächliche Verwaltungsausgaben stehen als Grundausstattung voraussichtlich zur Verfügung	Mittel für sächliche Verwaltungsausgaben in der Grundausstattung werden zentral vom Fachbereich Elektro- und Informationstechnik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden.		
Für sächliche Verwaltungsausgaben werden als Ergänzungsausstattung beantragt (entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7)	19.000	19.000	12.000

Begründung zur Ergänzungsausstattung der sächlichen Verwaltungsausgaben

Jedem der Mitarbeiter und studentischen Hilfskräfte steht ein aus der Grundausstattung finanzierter Arbeitsplatz PC zur Verfügung. Dieser dient zur Projektbegleitung und Programmentwicklung.

Zur Entwicklung höherer Regelungsstrategien sind eine Reihe sogenannter *Control-Design-Tools* (Matlab/Simulink-Control-Toolboxes) kommerziell verfügbar und sollen beschafft werden. Zur adäquaten Behandlung regelungstechnischer Problemstellung ist deren Nutzung unbedingt erforderlich. Hinzu kommen die Kosten für regelmäßige Updates der Matlab/Simulink aber auch der Dymola Simulationsumgebung. Auch ist die Anschaffung zusätzlicher Lizenzen für die studentischen Mitarbeiter absolut notwendig.

Zur simulativen Validierung verfügt das Teilprojekt über erprobte, *kommerzielle Gebäudesimulatoren*. Um mit aktuellen Entwicklungen Schritt zu halten, sind auch hier regelmäßige Updates notwendig. Hierzu zählt zunächst der Übergang zur nun C-basierten und auch modelltechnisch weiterentwickelten *Release 15.0* des Gebäudesimulators TRNSYS.

Zur messtechnischen Validierung der Modelle und Algorithmen ist eine Erweiterung der bisherigen technischen Hardwareausstattung erforderlich. Gerade die Anbindung

an moderne Simulationstools erfordert die Anschaffung passender Interface-Lösungen. Hierzu existiert aus heutiger Sicht eine für regelungstechnische Fragestellungen gut geeignete, zu Matlab/Simulink kompatible, kommerzielle Hardware-in-the-Loop-Lösung (*xPC target 1.0.2*).

Ausgangspunkt ist die Matlab/Simulink-Simulationsoberfläche.

Zunächst sind die in Dymola/Modelica entworfenen Streckenmodelle bzw. die Algorithmen in die Matlab/Simulink-Simulationsoberfläche als sogenannte Dymolablöcke einzubinden.

Das Softwarepaket *xPC-Target* erlaubt die unmittelbare Integration und Nutzung von Ein/Ausgabeblocken unter Matlab/Simulink, die echte Schnittstellen zur realen Welt repräsentieren.

Beide werden unter Matlab/Simulink, eventuell unter Verwendung weiterer notwendiger Simulinkblöcke zu einem Gesamtmodell des Prototypes verbunden.

Unter Anwendung der Matlab/Simulink eigenen Toolbox (*Real-time-Workshop*) kann dann durch automatische Codegenerierung eine prototypische Implementierung für den eigentlichen *Mess- und Steuerungs-PC* erzeugt werden.

Auf dem *Mess- und Steuerungs-PC* läuft unter Anwendung des *xPC-Target-Softwarepaketes* in Echtzeit der Algorithmenprototyp und interagiert über geeignete *I/O-Karten* mit den benötigten *Sensoren* bzw. *Aktoren*.

Wesentliche Voraussetzung und ausschlaggebend für die Auswahl dieser Hardware-in-the-Loop-Lösung ist neben der Transparenz der Vorgehensweise die kurze Zykluszeit zwischen Änderung des Algorithmus auf abstrakter Modellebene und experimenteller Auswirkung.

Die Auswahl der neuen Testumgebungen kann erst später erfolgen, ihre aktuelle Verfügbarkeit berücksichtigend. Es sollen Testumgebungen ausgewählt werden, bei denen auch ein Zugriff auf die Aktorik möglich ist. Wesentlich ist, dass die Gebäude auch einen realen Heizungsbedarf aufweisen und für aktiv beeinflussende experimentelle Studien zur Verfügung stehen.

Da die Validierungen schon in AP-A2 beginnen und projektbegleitend weiterzuführen sind, sollen die wesentlichen Anschaffungen in 2001 erfolgen. Die Ausgaben in 2002 beziehen sich auf zusätzliche Erweiterungen und Instandhaltung. Für die ersten Tests steht ein geeigneter PC aus der Grundausstattung zur Verfügung. Sobald jedoch die Auswahl der Testumgebung vollzogen wurde und die Messeinrichtungen fest installiert werden, sollte ein ausschließlich für diese Experimente reservierter, zusätzlicher PC angeschafft werden.

Weitere sächliche Verwaltungskosten in der Ergänzungsausstattung werden zur Finanzierung von Literaturrecherchen, kleineren Geräteerweiterungen und elektronischem Kleinmaterial benötigt.

	2001	2002	2003
1 <i>Control Design Tools</i>	6 000.-	4 000.-	4 000.-
1 <i>kommerzielle Gebäudesimulatoren</i>	3 000.-	3 000.-	3 000.-
1 <i>Realtime Workshop</i>	2 000.-		
1 <i>Mess- und Steuerungs-PC</i>		4 000.-	
1 <i>PC Target 1.0-Softwarepaket</i>	2 000.-		
1 <i>DA/AD-Wandlerkarten</i>	2 000.-	3 000.-	
1 <i>Aktorik und Sensorik</i>	2 000.-	3 000.-	3 000.-
weitere sächliche Verwaltungskosten	2 000.-	2 000.-	2 000.-

3.7.3 Investitionen (Geräte über 20.000 brutto)

Bezeichnung des Gerätes (ggf. mit Typenbezeichnung)	Einzelpreis 2001	Einzelpreis 2002	Einzelpreis 2003
	-	-	-
Summe	-	-	-

4 Geschäftsordnung des SFB 501

- Entwicklung großer Systeme mit generischen Methoden -

4.1 Definition und Zweck des Sonderforschungsbereichs

4.1.1 Der SFB "Entwicklung großer Systeme mit generischen Methoden" hat sich am 8.10.1993 in Kaiserslautern konstituiert.

4.1.2 Der SFB ist ein institutionalisierter, d.h. durch diese Geschäftsordnung gebundener Zusammenschluß wissenschaftlicher Forschungsgruppen. Eine Forschungsgruppe wird durch freiwilligen Zusammenschluß mehrerer Forscher, die gemeinsam ein Forschungsprojekt oder mehrere eng verwandte Forschungsprojekte verfolgen wollen, gebildet.

Der SFB steht weiteren Interessenten offen, die ein Arbeitsprogramm gemäß den Richtlinien des SFB besitzen und zur Kooperation bereit sind.

4.1.3 Der SFB befaßt sich mit Methoden und Verfahren zur Verbesserung des Herstellungsprozesses für große komplexe Softwaresysteme. Durch generische Ansätze soll die Wiederverwendung von Produkten, Entwicklungsschritten und gesammelten Erfahrungen systematisch unterstützt werden.

4.1.4 Die Ziele des SFB sollen auf folgenden Wegen erreicht werden:

- Anregung, Koordination und Durchführung von Forschungsvorhaben verschiedener Forschungsgruppen.
- Organisation gemeinsamer Veranstaltungen, die dem Erfahrungsaustausch - auch auf internationaler Ebene - sowie der Darlegung der in den einzelnen Gruppen gewonnenen Forschungsergebnisse dienen.
- Finanzielle Planung von Forschungsvorhaben.
- Beteiligung in- und ausländischer Wissenschaftler am gemeinsamen Forschungsprogramm, gegebenenfalls durch Einrichtung neuer Stellen (z.B. Forschungsprofessuren) aus Mitteln des SFB.
- Beteiligung in- und ausländischer Wissenschaftler am gemeinsamen Forschungsprogramm, ggf. durch Einrichtung neuer Stellen (z.B. Forschungsprofessuren) aus Mitteln des SFB.

4.1.5 Der SFB stellt jährlich einen Tätigkeitsbericht zusammen.

4.2 Zugehörigkeit zum SFB

- 4.2.1** Die Forschungsgruppen bestehen aus Mitgliedern, Mitarbeitern sowie ggf. vorübergehend im SFB tätigen Wissenschaftlern anderer Forschungsstellen (s. Ziff. 4.1.4)
- 4.2.2** Die Mitgliedschaft im SFB steht Personen mit abgeschlossenem Hochschulstudium (Diplom, 1. Staatsexamen) offen, die auf dem in Ziff. 4.1.3 genannten Gebiet tätig sind und sich durch einschlägige wissenschaftliche Publikationen ausgewiesen haben.
- 4.2.3** Die Mitgliedschaft im SFB ist schriftlich beim Vorstand zu beantragen. Der Antrag muß durch Empfehlungen zweier Mitglieder des SFB oder durch Gutachten zweier nicht zum SFB gehörenden Experten unterstützt werden. Über die Aufnahme entscheidet die Mitgliederversammlung auf Vorschlag des Vorstandes mit einfacher Mehrheit. Nur Mitglieder können Anträge auf Teilprojekte stellen.
- 4.2.4** Die Mitgliedschaft im SFB endet:
- auf eigenen Wunsch des Mitglieds;
 - ein Jahr nach Einstellung aller Forschungsprojekte des SFB, an denen das Mitglied beteiligt war;
 - wenn das Mitglied seine Tätigkeit an der Universität Kaiserslautern aufgibt;
 - auf Antrag von mindestens 5 Mitgliedern und Beschluß der Mitgliederversammlung; der Beschluß muß mit 2/3 Mehrheit aller Mitglieder gefaßt werden.
- 4.2.5** Ausscheidende Mitglieder, die auf dem in Ziffer 4.1.3 genannten Gebiet weiter tätig und an einem Informationsaustausch interessiert sind, können auf Antrag "korrespondierendes Mitglied" des SFB (ohne Antrags- und Stimmrecht) werden. Über den Antrag entscheidet die Mitgliederversammlung mit Mehrheit. Für das Ausscheiden korrespondierender Mitglieder gilt die Bestimmung 4.2.4 sinngemäß.
- 4.2.6** Die Mitgliedschaft im SFB gewährt keinen Anspruch auf Mittelzuweisung.
- 4.2.7** Mitarbeiter sind Angehörige des SFB, die nicht Mitglieder sind. Über die Besetzung von Mitarbeiterstellen verfügt(en) der(die) zuständige(n) Projektleiter; über die Besetzung nicht-gruppen-spezifischer Mitarbeiterstellen entscheidet der Vorstand.

4.3 Mitgliederversammlung

- 4.3.1** Alle Mitglieder des SFB bilden die Mitgliederversammlung, die mindestens einmal pro Jahr zusammentrifft. Die Mitgliederversammlung ist darüber hinaus einzuberufen, wenn 1/5 der Mitglieder dies verlangen.
- 4.3.2** Die Mitgliederversammlung wird vom Sprecher einberufen.
- 4.3.3** Sie ist beschlußfähig, wenn mindestens die Hälfte aller Mitglieder anwesend sind.
- 4.3.4** Die Mitgliederversammlung nimmt den Bericht des Sprechers des SFB entgegen und nimmt ggfs. zu ihm Stellung.
- 4.3.5** Die Mitgliederversammlung kann Stellungnahmen zu den vom Vorstand ausgelegten Anträgen abgeben.
- 4.3.6** Sie entscheidet über die Aufnahme von Neumitgliedern.

4.4 Vorstand und Sprecher

- 4.4.1** Der Vorstand besteht aus den Teilprojektleitern.
- 4.4.2** Der Vorstand wählt in geheimer Wahl den Sprecher und den stellvertretenden Sprecher. Er schlägt der Mitgliederversammlung die Aufnahme neuer Mitglieder vor (s. 4.2.3) und verabschiedet den Tätigkeitsbericht. Der Vorstand wählt ständige oder "ad hoc" Kommissionen. Er entscheidet mit einfacher Mehrheit über alle Fragen, die die Organisation und Tätigkeit des SFB betreffen und die nicht nach dieser Ordnung anderen Gremien zugeordnet sind.
- 4.4.3** Die Amtszeit des Sprechers beträgt drei Jahre (Förderungsperiode).
- 4.4.4** Der Vorstand nimmt folgende Aufgaben wahr:
- Er entscheidet über die Aufnahme, Ablehnung und Beendigung von Projektanträgen unter Berücksichtigung der Stellungnahme der Mitgliederversammlung. Er muß seine Entscheidung begründen.
 - Er hat das Recht und die Pflicht, über die Verteilung der Geld-, Sach- und Personalmittel nach dem Gesichtspunkt des bestmöglichen Einsatzes zu verfügen, sofern durch diese Verfügung keine Verfälschung der Gutachterentscheidung bedingt wird.
 - Er organisiert gemeinsame Veranstaltungen.

4.4.5 Der Sprecher hat folgende Aufgaben:

- Er vertritt den SFB nach außen.
- Er beruft und leitet die Sitzungen des Vorstandes und der Mitgliederversammlung. Er ist ex officio Mitglied aller Kommissionen.
- Er stellt die Gesamtanträge bei der Deutschen Forschungsgemeinschaft.
- Er faßt die Jahresberichte der Projektleiter zu einem Gesamtbericht zusammen und redigiert sie ggf.
- Er berichtet der Mitgliederversammlung über die Tätigkeit des Vorstandes und über die Anträge auf Mitgliedschaft und Projektaufnahme.
- Die Ladung zur Mitgliederversammlung ergeht 14 Tage vor dem Termin der Versammlung. Sie enthält eine vollständige Tagesordnung. Während der Ladefrist liegen alle Neuanträge auf Teilprojekte aus.

4.5 Verfahren für die Ausarbeitung der Anträge und die Mittelbewirtschaftung

4.5.1 Anträge auf Mittelzuweisung im Rahmen des SFB können nur von Projektleitern gestellt werden.

4.5.2 Der SFB unterscheidet zwei Typen von Anträgen an die DFG:

- A: Anträge von allgemeinem Interesse für den SFB, z.B. Anträge auf Großgeräte, zentrale Einrichtungen, Forschungsprofessuren, Unterstützung von Gastwissenschaftler-Aufenthalten und Reisekosten.
- B: Anträge auf Förderung spezifischer Forschungsprojekte eines oder mehrerer Projektleiter des SFB.

4.5.3 Alle Anträge gemäß Ziffer 5.2 werden vom Sprecher nach Verabschiedung durch den Vorstand zu einem bestimmten Termin des Jahres zusammengefaßt und an die Deutsche Forschungsgemeinschaft weitergereicht.

4.5.4 Alle Anträge gemäß Ziffer 5.2 müssen eine detaillierte Aufgabenstellung, Begründung und Finanzplanung enthalten. Die Finanzierung kann maximal für einen Zeitraum von drei Jahren geplant werden. Die Anträge können auch Zusatzanträge zu den bereits geförderten Vorhaben beinhalten, soweit diese Förderung nicht von der Deutschen Forschungsgemeinschaft außerhalb des SFB erfolgt.

4.6 Gemeinsame Projekte und Einrichtungen

- 4.6.1** Einrichtungen, die dem SFB als Ganzem zur Verfügung stehen, werden durch Entscheidung des Vorstandes einem Projektleiter des SFB verantwortlich zugeordnet.
- 4.6.2** Beim Ausscheiden eines Projektleiters aus dem SFB können die aus Mitteln des SFB erworbenen Geräte, Materialien und andere Forschungshilfen durch Vorstandsentscheid einem anderen Projektleiter für Arbeiten im Rahmen des Sonderforschungsbereichs übergeben werden. Wenn es mit den Interessen des SFB vereinbar ist, kann der Vorstand im Einzelfall beschließen, die Geräte, Materialien und andere Forschungshilfen dem früheren Mitglied ggf. befristet zu belassen.

4.7 Schlußbestimmungen

- 4.7.1** Diese Ordnung ist mit der Konstituierung des SFB 501 am 8.10.1993 in Kraft getreten.
- 4.7.2** Die Änderung der Ordnung kann nur mit Zweidrittelmehrheit der Mitgliederversammlung durchgeführt werden.