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 Verwaltungsausgaben	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 Artifakte, 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 entwikkelt 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. Simulationspattern). 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 ähnlich 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 einen 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://wwwagz.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 einen 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 Artifakte 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 Artifakte 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, Klimatisierung, 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.*

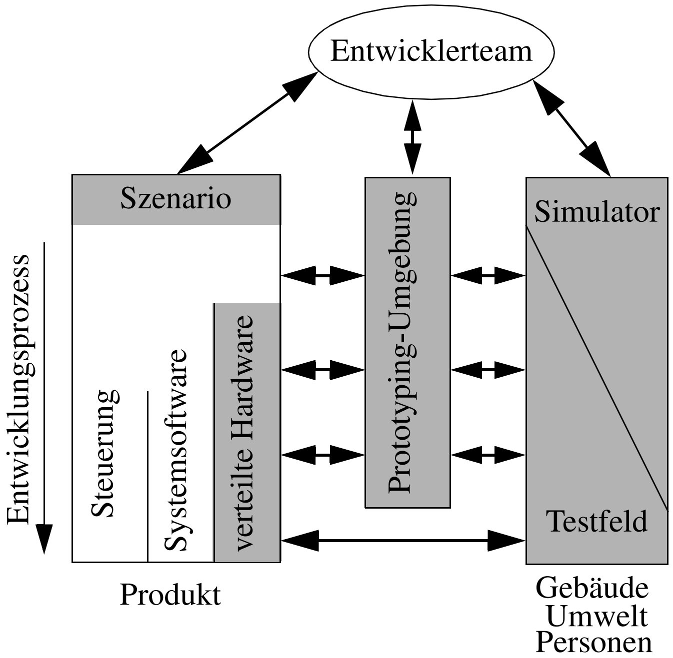

Abbildung 1: Testumgebung

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 Artifakte 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 Analysemethode 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 Modellierungselemente (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 objektorientierten 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

Gebäudeautomationssytem 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.*

Als Teil der Prototyping-Umgebung haben wir die Schnittstellen zum Testfeld, zu Gebäudesimulatoren und zu dem zu testenden Automationssystem realisiert. Die Komponenten des Testfeldes sind über CAN und Ethernet ansprechbar. Als Simulatoren wurden bisher ein nach der Spezifikation von Team 1 von Teilprojekt C1 in Erlang geschriebener Simulator und der Beleuchtungssimulator Lumina [8] von der CMU adaptiert. Der Team1-Simulator verfügt über eine Socket-Schnittstelle, wohingegen die Kommunikation mit Lumina über CORBA erfolgt [9]. Kontrollsystem-Prototypen waren entweder in C++ handkodiert (CS2) oder aus SDL-Modellen mit SDT generiert (CS3/1 [21] und CS3/2 [23]). Für diese Fallstudien hat sich die Prototyping-Umgebung sehr gut bewährt und ist für die Analysephase ausreichend.

In der nächsten Phase soll das Prototyping vor allem für diejenigen Entwicklungsmethoden ausgebaut werden, die in den Teilprojekten B4, B5 und B11 entwickelt werden. Besondere Anforderungen ergeben sich aus der gleichzeitigen Entwicklung von Hard- und Software, wie auch in Abb. 1 gezeigt. Wir nennen diese Technik deshalb *Hardware-Software-Co-Prototyping*. Das entwurfsbegeleitende Protoyping muss es ermöglichen, für jeden Entwicklungsstand und jede Konfiguration die getroffenen Entscheidungen überprüfen und bewerten zu können. Durch geeignete Testfälle und Fragestellungen müssen dabei vor allem nichtfunktionale Anforderungen überprüft werden, wenn man davon ausgeht, dass die in der Analysephase überprüften funktionalen Eigenschaften während des Designs nicht verfälscht werden. Zusätzlich muss das konkurrente Verhalten überprüft werden, wenn Prozesse während des Designs zusammengelegt oder aufgeteilt werden. Vor allem wollen wir uns in Zusammenarbeit mit B11 auf die Fehlertoleranz konzentrieren und für das Realzeitverhalten Lösungen anderer Forschungsgruppen (z.B. Färber [10]) übernehmen.

Co-Prototyping erfordert auch die teilweise Installation der Produktsoftware auf der verteilten Hardware, um zum Beispiel Anforderungen an das Kommunikationssystem ableiten oder überprüfen zu können. Da das Testfeld keine beliebigen Szenarien liefern kann (z.B. ist während der Sommermonate ein Heizbetrieb ausgeschlossen) und nicht beliebig skalierbar ist, müssen daher auch die Simulation oder zumindest die Simulationsdaten verteilt werden, um „künstliche" Stimuli für das System zu erzeugen. Dazu müssen Methoden entwickelt werden, die bei allen Konfigurationen eines *verteilten Prototyping* die Skalierbarkeit der Ressourcen und die Synchronisation gewährleisten, um die Aussagen nicht zu verfälschen. Es müssen Möglichkeiten im Testfeld und in den Simulatoren geschaffen werden, um die Überprüfung der Fehlertoleranz durch die gezielte Simulation von Fehlern zu ermöglichen.

Wie bei allen Methoden im Zusammenhang mit der Entwicklung von großen Systemen spielt die *Effizienz* eine Rolle. Das Prototyping muss deshalb fest in den Prozess eingebunden werden und der Nutzen, z.B. eine Qualitätsverbesserung des Produktes, muss in einem vernünftigen Verhältnis zu dem Zusatzaufwand stehen. Daher müssen zeitaufwendige Schritte im Prototyping-Prozess durch den Einsatz von Werkzeugen unterstützt werden.

Methode: Die Erzeugung von partiellen, bzw. verteilten Prototypen wird durch Werkzeuge der Prototyping-Umgebung ermöglicht. Ausgangspunkt für die Generierung von Prototypen bildet die textuelle Darstellung der SDL-Modelle (SDL-PR). Nach dem Parsen werden diese Modelle werkzeugunterstützt um Treiber und Stubs, die notwendigen SDL-Fragmente für die Kommunikation zwischen den Prototyp-Partitionen und eine Instrumentierung für den Nachweis von konkurrenten und nichtfunktionalen Eigenschaften ergänzt. Zusätzlich werden die Modelle so erweitert, dass auch Simulationsdaten (an Stelle von echt gemessenen Werten oder zur Simulation von Fehlern) im verteilten Prototyp verwendet werden können. Aus dem neu generierten SDL-Modell wird schließlich ausführbarer Code für unterschiedliche Zielsysteme erzeugt. Eine wichtige Fragestellung ist dabei die Synchronisierung der einzelnen Komponenten. Dies wirft ähnliche Probleme wie die Kopplung von Simulatoren auf, weshalb hier Lösungen aus diesem Bereich eingesetzt werden. Die benötigten Werkzeuge sollen iterativ und möglichst aus schon existierenden Werkzeugen entwickelt oder durch Integration mit anderen Tools erstellt werden. So verwenden wir z.B. für das Erzeugen von C-Code aus SDL-PR-Dokumenten den Code-Generator von SDT.

Die Effizienz von Prototyping wird durch den Einsatz von Werkzeugen und das Ausnutzen domänenspezifischer Eigenschaften erreicht.

Nachweis: Die Realisierbarkeit der Werkzeuge wird durch deren Implementierung und die Einbindung in die Prototyping-Umgebung nachgewiesen. Die Funktionsfähigkeit muss durch anschließenden Fallstudien, bei denen die erweiterte Prototyping-Umgebung eingesetzt wird, validiert werden.

Es wird in Experimenten gemessen, wie groß der Nutzen (Qualitätsverbesserung, Zeitersparnis) des Einsatzes von Prototypen bei der Requirements Analyse und beim

Design im Vergleich zum Aufwand ist und in wie weit die Spezialisierung auf eine Domäne Verbesserungen bringt.

Schwerpunkt D: Requirements Engineering

In dem vom SFB bearbeiteten Gesamtprozess zur Entwicklung großer reaktiver Systeme beschäftigt sich D1 mit der Requirements-Engineering-Phase (auch Analysephase). Dabei hat D1 es sich zur Aufgabe gemacht, diese wichtige Phase möglichst effizient zu gestalten, so dass sie sowohl in Experimenten als auch in der Praxis schnell und zuverlässig durchgeführt werden kann.

Vision: *Durch eine effiziente Methode soll der Entwickler zusammen mit dem Kunden und den Domänenexperten so unterstützt werden, dass aus vagen, unvollständigen Kundenanforderungen formale, vollständige System-Requirements entstehen. Diese sollen alle geforderten und während dieser Phase hinzugekommenen funktionalen und nichtfunktionalen Anforderungen erfüllen oder als Constraints propagieren. Der in dieser Analysephase investierte Aufwand muss sich für den Gesamtentwicklungsprozess des Produkts und der Produktvariantenbildung bezüglich Qualität oder Kosten auszahlen.*

Erreichter Stand: Ausgehend von der hierarchischen Gebäudestruktur wurde eine Modellarchitektur entwickelt, die die Steuerungsfunktionen hierarchisch auf Prozesse verteilt und dadurch Regularität, Wiederverwendung, Gruppenarbeit, Verfolgbarkeit und Prototyping von Teilmodellen effizient unterstützt und die direkt in SDL umsetzbar ist. Solche Strukturen existieren in vielen Domänen. Daraus ergab sich ein einfaches Produktmodell und ein auf Verfeinerung basierender Analyseprozess, der durch einfache Modell-Pattern und durch Domänenwissen aus einem Dictionary unterstützt wird. Eine einfache Prozessumgebung mit HTML-Dokumenten und Perl-Skripten automatisiert Versionierung und Aufwandsdatenerfassung. Durch eine Reihe von Fallstudien wurde nachgewiesen, dass dieser Prozess auch bei größeren Problemstellungen von einem Team in sinnvoller Zeit System-Requirements hoher Qualität erzeugt, die durch Inspektion und Prototyping validiert und verifiziert wurden und für die weiteren Phasen zur Verfügung stehen (Ergebnisbericht). Durch ein auf Traces während des Prototyping basierendes Analysewerkzeug stehen auch dynamische Leistungsdaten der System-Requirements zur Verfügung. Ergänzt wird dieses Vorgehen durch die in der Kooperation B4/C1 entwickelte Formalisierung der Problembeschreibung [18]. Diese Formalisierung kann in unsere Methode integriert werden.

Ziel 1: Nachdem jetzt konkrete Erfahrungen mit dem Analyseprozess vorliegen, soll als Verallgemeinerung eine *Entwurfssystematik* für diese Phase entwickelt werden, die auch auf andere Phasen des Prozesses erweitert werden kann. Mit dieser Systematik sollen Produkt-, Prozess- und Vorgehensmodelle auf eine Basis gestellt werden, die es erlaubt, Attribute wie Verfeinerungsgrad, Entwurfszustand und Abhängigkeiten wie Verfolgbarkeit, Benachrichtigung, Version-von und Alternative-von klar zu definieren. Während die Behandlung funktionaler Eigenschaften relativ klar ist, stellt die Integration nichtfunktionaler Eigenschaften ein schwieriges Problem dar. Dasselbe gilt für das vor allem während des Requirements-Engineering einfließende Expertenwissen. Wir suchen nach einer Lösung, auch diese Komponenten in der Entwurfssystematik und damit in die genannten Modellen richtig einzuordnen. Desweiteren soll der Zusammenhang zwischen Constraints (z.B. Zuverlässigkeits- oder Zeitanforderungen) und Lösungen für diese betrachtet werden. Dies ist insofern problematisch, da sowohl Lösungen schon vom Kunden vorgegeben werden können und diese dann als Constraints in die Entwicklung einfließen, als auch die in einer Entwicklungsphase gefundenen Lösungen in späteren Phasen wieder revidiert werden können. In diesem Fall würden die in einer früheren Entwicklungsphase gefundenen Lösungen in einer späteren Phase sowohl als Constraints als auch als potentielle Lösungen betrachtet werden können.

Methode: Es werden zunächst die kleinsten versionierten Artifakte als Features definiert, die durch Elementaroperationen bearbeitet oder verknüpft werden. Durch Konfigurationen von Feature-Versionen entstehen zusammengesetzte Artifakte, die über eine Repräsentation Dokumente definieren. Zusammengesetzte Elementaroperationen definieren Prozess-Schritte. Die Kunst liegt vor allem darin, die Features und Konfigurationen so zu definieren, dass erstens einfache Modelle (insbesondere das Vorgehensmodell) entstehen, die für den Entwerfer leicht verständlich und nachvollziehbar sind, dass zweitens redundante Information und die Ausbreitung von Änderungen minimiert werden und dass drittens die Konsistenz aller Artifakte eines Produktes leicht gewahrt werden kann. Für die Betrachtung des Zusammenhangs zwischen Constraints und Lösungen eines Entwicklungsschrittes muss zunächst ermittelt werden, welche Lösungen in diesen Phasen noch Freiheitsgrade aufweisen, bzw. in welchen Phasen die Lösungen unveränderbar festgelegt werden. Diese Erfahrungen können dann verallgemeinert in die Entwurfssystematik einfliessen.

Für die Anwendung der Systematik sollen Werkzeuge prototypisch gebaut werden, die auf den verschiedenen Modellen (Produkt-, Prozess- und Vorgehensmodell) arbeiten. Diese Werkzeuge sind notwendig, um Experimente, mit deren Hilfe die theoretischen Überlegungen bewertet werden sollen, sinnvoll durchführen zu können.

Ziel 2: Auch wenn die durchgeführten Experimente bereits einen hohen Grad an Effizienz, Produktqualität und Produktvollständigkeit zeigen, sind weitere Verbesserungen notwendig. Dementsprechend ist ein weiteres Ziel, die *Analysemethode* weiterzuentwickeln, um z.B. die System-Requirements in Bezug auf nichtfunktionale Eigenschaften zu erweitern. Dazu sollen zusätzliche Modellarchitekturen untersucht und andere Notationen, wie zum Beispiel MSCs, hinzugenommen werden, soweit sie den Modellierungsaufwand verringern oder das Produkt verbessern. Solche Änderungen sollen auf der Basis der Entwurfssystematik in das Produktmodell und den Prozess integriert werden. Reuse von Domänenwissen und Artefakten soll durch eine enge Verknüpfung von Projektdaten, Dictionary und Reuse-Library durch ein gemeinsames Datenmodell erfolgen und bewertet werden.

Methode: Erweiterungen sollen nach Möglichkeit auf bekannten Modellierungstechniken beruhen, für die auch Werkzeuge existieren. Für Echtzeitanforderungen sind das zum Beispiel SDL-Annotationen, wie sie in dem REAR-Framework [10] verwendet werden. Jede Erweiterung muss zunächst in den Prozess integriert werden und durch Experimente bewertet werden. Außer dem Aufwand soll vor allem auch die Produktqualität betrachtet werden. Qualität bedeutet hier außer Korrektheit auch Detaillierungsgrad, Vollständigkeit und Informationen, die die folgenden Entwicklungsphasen vereinfachen. Dazu werden zum Beispiel mit Hilfe einer dynamischen Analyse Zeitanforderungen für Sensor-Aktuator-Paare ermittelt und propagiert. Andererseits kommen aus diesen Phasen auch Informationen zurück, durch die das Produkt der Analysephase verfeinert wird. Mit dem Fortschritt in Teilprojekt B11 soll dann dasselbe auch für Zuverlässigkeitsanforderungen durchgeführt werden, die einen ganz anderen Charakter als Zeitanforderungen haben. Bewertet werden die einzelnen Erweiterungen wiederum durch Experimente. Alle erfolgreichen Erweiterungen werden in die Entwurfssystematik eingebracht.

3.5.2 Arbeitsprogramm

Arbeitsschwerpunkt A: Testfeld

Arbeitspaket A1: Integration der Simulation

Um die Skalierung zu großen Gebäudestrukturen durch Intergration von Simulation und Testfeld korrekt durchzuführen, müssen an allen Stellen, an denen die Simulation das Testfeld beeinflusst, entsprechende Maßnahmen getroffen werden, so dass sich das Testfeld in Bezug auf das getestete System weitgehend korrekt verhält. Dazu werden zum Beispiel Nachbarräume zu den Testfeldräumen so ausgestattet, dass sie das von der Simulation berechnete Temperaturverhalten emulieren, um die korrekten Wärmeflüsse zum Testfeldraum zu erzeugen. Ähnliche Lösungen müssen zum Beispiel für die Wetter- oder Personensimulation gefunden werden. Die Beeinflussung der Simulation durch das Testfeld erfolgt durch zusätzliche Sensoren. Für diese Kopplungskomponenten wird ein von der Steuerung unabhängiges Kommunikationssystem aufgebaut.
Personalbedarf: 18 PM

Arbeitspaket A2: Kommunikationsstruktur

Die flexible Vernetzung wird durch zusätzliche Netzwerkkomponenten realisiert. Zusätzliche CAN- und Ethernet-Busse werden zwischen den Prozessorknoten installiert. Dabei kann für die Ethernet-Verkabelung und teilweise auch für langsame CAN-Busse auf vorhandene Infrastruktur zurückgegriffen werden. Für die flexible Kommunikationsstruktur werden Routing- und Protokoll-Strategien, die in B4 entwickelt wurden, auf die Zielhardware gebracht. Es erfolgt eine Trennung der „echten" Kommunikation zwischen Knoten von der „Prototyping"-Kommunikation durch Verwendung unabhängiger Kommunikationsmedien (z.B. CAN und Ethernet). Damit ist auch eine verteilte Simulation möglich, die zunächst nur durch eine Verteilung der simulierten Sensoren und Aktuatoren realisiert wird.
Personalbedarf: 18 PM

Arbeitsschwerpunkt B: Simulator-Entwicklungsmethode

Arbeitspaket B1: Simulatorarchitekturen

In diesen Arbeitspaket soll untersucht werden, wie die Modellierung gewünschter Simulatoreigenschaften durch vorgegebene Architekturen oder Architekturstile optimal unterstützt werden kann. Insbesondere sollen dabei die Aspekte Verteilung und Modellintegration betrachtet werden. Zur Verteilung muss es einfach möglich sein, die Struktur des Simulators zu partitionieren und Kommunikationsmechanismen (z. B. Pipes&Filters, CORBA) an den Schnittstellen einzufügen. Verschiedene Modelle (z. B. CAD, Klassendiagramme, Endliche Automaten) ermöglichen es einem Entwerfer, die einzelnen Aspekte des Simulators passend modellieren zu können. Diese Teilmodelle müssen durch Transformationen in ein widerspruchsfreies und möglichst vollständiges Generatormodell überführt werden, aus dem der Simulator-Code automatisch erstellt werden kann. Dazu soll auch SDL als Modellierungssprache erprobt werden.
Personalbedarf: 21PM

Arbeitspaket B2: Einbettung in den Anwendungssystem-Entwicklungsprozeß

Um effizient Simulatoren erzeugen zu können, muss eine geeignete Modellierungsmethode aufgestellt werden. Diese soll es ermöglichen, parallel zum Entwurf des Steuerungssystems einen Simulator zu modellieren, mit dem die zu überprüfenden Szenarien ablauffähig sind. Dabei soll viel von den bisher aufgestellten Prozessmodellen benutzt und um simulationsspezifische Details erweitert werden. Zusätzlich müssen auch Schnittstellen zwischen Steuerungssystem, Testraum und Simulator definiert werden. Es soll möglich sein, Simulatoren parallel zum Steuerungssystem zu entwerfen. Daher sollten sich Simulator und Steuerungssystem auf gleichem Abstraktionsniveau befinden und vorhandene Gemeinsamkeiten etwa in Form von gleichen Basismodellen miteinander teilen.
Personalbedarf: 12PM

Arbeitspaket B3: Überprüfung

Die Überprüfung unserer Simulator-Entwicklungsmethode gliedert sich in zwei Teilpunkte auf: Zum einen muss mit Software-Engineering Experimenten die Tauglichkeit der Entwurfsmethode nachgewiesen werden und zum anderen muss die Qualität der modellierten und generierten Simulatoren ermittelt werden. Für beide Aspekte sollen zahlreiche Experimente geplant und durchgeführt werden. Die Genauigkeit der Simulatoren kann durch Vergleichsmessungen im Testraum und im Vergleich zu anderen Simulatoren festgestellt werden.
Personalbedarf: 12PM

Arbeitsschwerpunkt C: Prototyping

Arbeitspaket C1: Partielles und verteiltes Prototyping

In diesem Arbeitspaket wird zunächst ein Prototyping-Werkzeug entwickelt, das die Generierung von partiellen Prototypen durch den Einsatz von Treibern und Stubs erlaubt. Dazu muß eine Methode gefunden werden, diese Treiber und Stubs aus den vorhandenen informellen und formalen Beschreibungen der zu ersetzenden Teilgraphen der Gesamtstruktur zu generieren. Dazu soll ein Wiederverwendungsansatz mit Pattern und generischen SDL-Modellen aus einer Bibliothek entwickelt werden. Die Implementierung des Werkzeuges erfolgt in Java und für die Kommunikation zwischen dem Prototyp und der Umgebung werden zunächst Sockets eingesetzt. Anschließend wird das Werkzeug um die Möglichkeit der Partitionierung des Gesamtsystems erweitert und andere Wege der Kommunikation zwischen den Prototypen und der Umgebung eingesetzt. Der sinnvolle Einsatz verschiedener Kommunikationslösungen auf unterschiedlichen Abstraktionsebenen (TCP/IP, QNX-Networking, Sockets, JavaRMI, CORBA) wird untersucht.
Personalbedarf: 8PM

Arbeitspaket C2: Hardware-Software-Co-Prototyping

Hier erfolgt die Erweiterung des Prototyping-Werkzeugs, um eine Instrumentierung der Prototypen durchführen zu können. Dies soll auf Modell-Ebene durch Einfügen von SDL-Fragmenten oder auf Code-Ebene durch Einbinden von C-Funktionsaufrufen erfolgen. Mit Hilfe der so instrumentierten Prototypen soll untersucht werden, wie sich nichtfunktionale und konkurrente Eigenschaften mit Hilfe von Prototypen nachweisen lassen. Hierzu werden sowohl neue Experimente durchgeführt, als auch SDL-Modelle existierender Fallstudien verwendet.

Für das Problem der Synchronisation der Prototyp-Partitionen untereinander und mit dem Testfeld sollen existierende Lösungen, die bei einer verteilten Simulation eingesetzt werden, evaluiert werden und geeignete Lösungsansätze in das Prototyping-Werkzeug integriert werden.

Zusätzlich werden Schnittstellen definiert, die es erlauben über die vorhandene Kommunikationsstruktur, neben den echten Gebäudedaten auch „künstliche” Stimuli von Simulatoren an die Prototypen heranführen zu können. Damit in Zusammenarbeit mit B4 Messungen des Kommunikationsverhaltens möglich werden, darf die Übertragung dieser Stimuli keinen Einfluss auf die eigentliche Kommunikation des Systems zeigen. Die Implementierung dieser Schnittstelle für den einzelnen Prototyp soll durch das Prototyping-Werkzeug unterstützt werden.
Personalbedarf: 8PM

Arbeitspaket C3: Effizienz

Begleitend zu AP-C2 werden Experimente durchgeführt und Ergebnisse bereits abgeschlossener Fallstudien ausgewertet, um nachzuweisen, dass effizientes Prototyping durch die entwickelten Werkzeuge möglich ist. Die Effizienz wird möglichst dadurch gezeigt, dass Entwurfsschritte jeweils mit und ohne Prototyping durchgeführt werden und neben des Aufwands auch die Qualität der entstandenen Produkte verglichen wird.

Zusätzlich wird bewertet, in welchen Maße die Spezialisierung auf eine Domäne Effizienzverbesserungen bewirkt. Dazu werden Unterschiede zu anderen Domänen identifiziert und mögliche Auswirkungen auf das Prototyping analysiert.
Personalbedarf: 8PM

Arbeitsschwerpunkt D: Requirements-Engineering

Arbeitspaket D1: Entwurfssystematik

Zunächst werden die in der Analysephase bereits durchgeführten Experimente in Bezug auf die kleinsten Modellierungseinheiten und Elementaroperationen analysiert. Es muss evaluiert werden, welche der Teilprodukte als solche kleinsten Einheiten und welche als zusammengesetzte Artifakte betrachtet werden können. Daraus wird dann eine abstrakte Sicht des Produktdatenmodells gebildet. Dieses Modell wird dann um eine abstrakte Sicht auf die dynamischen Abläufe und die benötigten Abhängigkeiten zwischen den Produkten innerhalb der Analysephase erweitert. Insbesondere wird untersucht, wie sich aus dem Entwicklungszustand der Modellierungseinheiten der Zustand oder Verfeinerungsgrad des Produktes ableiten lässt. Schließlich wird untersucht, welche Versionierungstypen den Entwickler sinnvoll unterstützen. Neben der Analysephase werden die Eigenschaften anderer Entwicklungsphasen (insbesondere der Designphase aus B11) in diese Betrachtungen einfliessen.

Weiterführend werden am Beispiel der Analysephase zu betrachtende Constraints und ihre Lösungen ermittelt und eine Abstraktion von ihnen in das entstehende Modell eingebracht. Auch hier muss eine Beziehung zwischen dem Erfüllungsgrad der Constraints und dem Entwicklungszustand des Produktes hergestellt werden.

Bei allen Abstraktionen muss immer wieder überprüft werden, ob sich diese zu den betrachteten Phasen wieder spezialisieren lassen. Weiterhin werden die theoretischen Überlegungen durch Experimente, unterstützt durch Werkzeuge aus AP-D3, bewertet.
Personalbedarf: 18PM

Arbeitspaket D2: Analysemethode

In diesem Arbeitspaket wird die Analysemethode um benötigte Beschreibungstechniken erweitert. Begonnen wird mit der Einführung von MSCs. Dazu muss das Produktdatenmodell erweitert werden und die Bearbeitung und Verwendung der MSCs in den dynamischen Ablauf der Analysephase integriert werden. In gleicher Weise werden neue Techniken eingebracht, deren Bedarf aus anderen Phasen folgt. Weiterhin müssen die in AP-D1 gefundenen Ergebnisse bezüglich der Modellierung der Constraints und ihrer Lösungen in die Analysemethode integriert werden. Die explizite Darstellung von Expertenwissen (z.B. die Realisierung von Reglern) aus Teilprojekt D2 muss durch neue Notationen unterstützt werden.

Parallel dazu wird an der Integration von Dictionary mit dem Domänenwissen und einer Reuse-Bibliothek gearbeitet. Nach der Verknüpfung dieser beiden Komponenten untereinander wird der Zugriff explizit in die Methode involviert.

Desweiteren wird an einer Verbesserung der Bewertung für die Durchführung einer Analysephase gearbeitet. Dies betrifft die Bewertung der entstehenden Produkte und der Abschätzung des benötigten Aufwands (und Kalenderzeit) der einzelnen Entwurfsaktivitäten. Dabei wird insbesondere Wert auf die Abschätzung des Nutzens von Reuse-Aktivitäten gelegt.

Der Nutzen jeder Erweiterung und Verbesserung muss wiederum durch exemplarische Durchführung in Experimenten validiert werden.
Personalbedarf: 16PM

Arbeitspaket D3: Werkzeugentwicklung

Bei der Durchführung von Experimenten in AP-D1 und AP-D2 dürfen die Ergebnisse nicht durch zusätzlich verursachten Aufwand verfälscht werden. Deswegen müssen prototypisch Werkzeuge erstellt werden, die die zu überprüfenden Erweiterungen unterstützen. Es ist geplant, in Zusammenarbeit mit dem Teilprojekt A3 die umfassende Erfahrungsdatenbank um eine Komponente zu erweitern, in der die Produktdaten der Analysephase verwaltet werden. An diesen projektspezifischen Teil wird dann MILOS (Teilprojekt A2) angegliedert, um den dynamischen Ablauf eines Projekts zu unterstützen und die Prozessdaten (Aufwand, Kalenderzeit) zu verwalten.
Personalbedarf: 15PM

Arbeitspaket D4: Wissenspflege und Experimentunterstützung

In den bisherigen Förderperioden wurde Domänenwissen in Form eines globalen Lexikons als Teil der Erfahrungsdatenbank gesammelt. Dieses Wissen muss ständig um das in den Experimenten neu hinzugekommene oder verfeinerte Wissen erweitert werden, zum Beispiel um MATLAB- oder Modelica-Modelle für Regelung, Steuerung und Simulation. Solche Modelle entstehen in D2 und auch in D1. Sobald von A3 eine geeignete Datenbankunterstützung für die Erfahrungsdatenbank existiert, kann diese auch um dieses Lexikon erweitert werden. Für alle Experimente werden Anwendungsszenarien vorgegeben, die sich aus dem Testfeld, aber auch nach Ideen aus dem Intelligent Workplace ergeben.
Personalbedarf: 8PM

3.5.3 Zeitplan

<table>
<tr><th>Personal</th><th>2001
1. H.J.</th><th>2001
2. H.J.</th><th>2002
1. H.J.</th><th>2002
2. H.J.</th><th>2003
1. H.J.</th><th>2003
2. H.J.</th></tr>
<tr><td colspan="7">Grundausstattung</td></tr>
<tr><td rowspan="3">A. Metzger</td><td colspan="2">AP C1</td><td colspan="4"></td></tr>
<tr><td></td><td colspan="4">AP C2</td><td></td></tr>
<tr><td></td><td colspan="5">AP C3</td></tr>
<tr><td rowspan="3">N.N.</td><td colspan="6">AP A1 und A2 (Planung und Leitung)</td></tr>
<tr><td colspan="6">AP D1 und D4</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td rowspan="2">J. Heiderich,
Techniker</td><td colspan="6">AP A1 und A2 (HW-Entwicklung)</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td colspan="7">Ergänzungsausstattung</td></tr>
<tr><td rowspan="3">S. Queins</td><td colspan="6">AP D1</td></tr>
<tr><td colspan="4">AP D2</td><td colspan="2"></td></tr>
<tr><td colspan="3">AP D3</td><td colspan="3"></td></tr>
<tr><td rowspan="3">J.P. Riegel</td><td colspan="3">AP B1</td><td colspan="3"></td></tr>
<tr><td colspan="2"></td><td colspan="3">AP B2</td><td></td></tr>
<tr><td></td><td colspan="5">AP B3</td></tr>
<tr><td rowspan="2">N.N.,
stud. Hilfskraft</td><td colspan="6">AP A1 und A2 (SW-Entwicklung)</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td rowspan="2">M. Trapp,
stud. Hilfskraft</td><td colspan="6">B1 und D3</td></tr>
<tr><td colspan="6"></td></tr>
</table>

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Als Infrastrukturprojekt ist D1 automatisch mit allen anderen Teilprojekten verknüpft. Enge Beziehungen bestehen einerseits zu allen B-Gruppen, die Experimente in der realen Domäne durchführen wollen, also B4, B5, B10 und B11. Mit D2 wird vor allem Expertenwissen ausgetauscht und interpretiert. Mit A1, A2 und A3 bestehen enge Beziehungen bezüglich der Werkzeugintegration und Experimentdurchführung. Im Bereich Requirements-Engineering ergänzen wir uns mit den Arbeiten von B4 und C1, die sich mit Spezifikationen in temporaler Logik beschäftigen. Die Erfahrung von C1 und A3 brauchen wir bei der Formalisierung der Entwurfssystematik.

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	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.:	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)		20.000	(522)		15.000	(522)		10.000
	zusammen		20.000	zusammen		15.000	zusammen		10.000
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 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. Gesamtarbeitszeit (beratend: B)	im SFB tätig seit	derzeitige Einstufung und beantragte Vergütungsgruppe
Grundausstattung 3.7.1.1 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Zimmermann, Prof. Dr.	Techn. Inf.	FB Inf., Uni KL	5	95	
	(2) Metzger, Dipl. Inform.	Techn. Inf.	FB Inf., Uni KL	20	98	
	(3) NN.			20		
3.7.1.2 Techniker	(4) Heiderich, Techniker	Elektrotech.	FB Inf., Uni KL	20	95	
Ergänzungsausstattung 3.7.1.3 wiss. Mitarbeiter (einschl. StdHkr.)	(1) Queins, Dipl. Inform.	Techn. Inf.	FB Inf., Uni KL	38,5	95	Bat IIa
	(2) Riegel, Dipl. Inform.	Techn. Inf.	FB Inf., Uni KL	38,5	96	Bat IIa
	(3) N.N., StdHkr.	Techn. Inf.	FB Inf., Uni KL	19		StdHkr.
	(4) Trapp, StdHkr.	Techn. Inf.	FB Inf., Uni KL	19	00	StdHkr.
3.7.1.4 nichtwissenschaftl. Mitarbeiter						

Aufgabenbeschreibung für Mitarbeiter der Grundausstattung

Der erste Mitarbeiter ist für die Bearbeitung der Arbeitspakete C1, C2 und C3 vorgesehen. Der zweite Mitarbeiter ist für die Mitarbeit in den Arbeitspaketen A1, A2 und D1 und für die Bearbeitung von D4 vorgesehen. Der Techniker ist für die technische Unterstützung der Arbeitspakete A1 und A2 vorgesehen.

Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausstattung

Der erste Mitarbeiter ist für die Bearbeitung der Arbeitspakete D1, D2 und D3 vorgesehen. Der zweite Mitarbeiter ist für die Bearbeitung der Arbeitspakete B1, B2 und B3 vorgesehen. Die erste studentische Hilfskraft ist für die Softwareunterstützung der Arbeitspakete A1, A2 und C1 vorgesehen. Die zweite studentische Hilfskraft ist für die Softwareunterstützung der Arbeitspakete B1 und D3 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)	25.000	15.000	10.000

Begründung zur Ergänzungsausstattung der Sächlichen Verwaltungsausgaben

Die beantragten Mittel sind notwendig, um die in den Arbeitspaketen A1 und A2 geplanten Erweiterungen des Testfeldes für die geplanten Experimente des SFB in einer realen Gebäudeumgebung mit integrierter Simulation durchführen zu können. Die angegebenen Zahlen entsprechen jetzigen Angeboten und unseren Erfahrungen in den beiden letzten Förderungsperioden. Die Ausstattung wurde auf zwei Jahre verteilt, da zum einen die Arbeitzeit verteilt werden muss, zum anderen aber auch flexibel auf Anforderungen aus dem SFB und bei technischem Fortschritt reagiert werden kann. Nur die CAN-Module müssen auf Vorrat gekauft werden, da Modelländerungen den Kauf einer neuen Entwicklungsumgebung erforderlich machen könnten.

Für das Jahr 2003 wurde ein Pauschalbetrag eingesetzt, da soweit nicht überschaubar ist, wie sich die Experimente im SFB entwickeln. Wir erwarten, dass das Testfeld dann vor allem bezüglich des realen Testens von Fehlertoleranz zusätzlich ausgestattet werden muss, wenn z.B. die Teilprojekte B4, B5 oder B11 entsprechende Lösungen erproben wollen.

Komponenten für die Erweiterung des Testfeldes	Einzel-preis	2001		2002	
		#	Summe	#	Summe
Controller					
CANModul-592 (8051 kompatibel)	250,–	22	5.500,–		
Platine für CANModul-592	50,–	22	60,–		
Netzgerät (5V/12V)	200,–	3	600,–	3	600,–
Sensorik/Aktuatorik					
Temperatursensor	50,–	25	1.250,–	15	750,–
Bewegungsmelder	50,–			10	500,–
Helligkeitssensor	20,–	20	400,–	10	200,–
Tür- und Fensterkontakte	5,–			50	250,–
Ansteuerung für Heizkörperventile	60,–	10	600,–	10	600,–
Kleinteile und Kabel (pro Raum)	300,–	3	900,–	3	900,–
Embedded PCs					
MSM486 (CPU-Modul)	725,–	2	1.450,–	4	2.900,–
MSM104 (Ethernet-Karte)	370,–	2	740,–	4	1.480,–
MSMCAN (CAN-Karte)	275,–	4	110,–	8	2.200,–
Flash-Speicher	120,–	2	240,–	4	480,–
Summe			24.230,–		14.660,–

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 D2

3.1.1 Thema

Modellbasierte Entwicklung wiederverwendbarer Regelungsalgorithmen

3.1.2 Fachgebiet und Arbeitsrichtung

Elektro- und Informationstechnik,
Automatisierungstechnik, Regelungs- und Steuerungstechnik

3.1.3 Leiter

Prof. Dr.-Ing. habil. Lothar Litz
geb. 30.8.1949
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-4450
Fax: 0631/205-4462
litz@eit.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 Verwaltungsausgaben	Investitionen	Gesamt
1998	201.6	-	0	201.6
1999	206.4	9.0	0	215.4
2000	208.8	12.4	0	221.2
Zwischensumme	616.8	21.4	0	638.2
2001	249,6	19.0	0	268,6
2002	249,6	19.0	0	268,6
2003	249,6	12.0	0	261,6

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

3.2 Zusammenfassung

a) *Teilprojekteigene Arbeitsgebiete*

Die teilprojekteigenen Forschungsgegenstände konzentrieren sich auf die modellbasierte Entwicklung von Regelungs- und Steuerungsalgorithmen bzw. deren Entwurfsmethodik. Dies setzt eine mathematische Modellierung des dynamischen Verhaltens des zu beeinflussenden technischen Prozesses, der sogenannten Strecke voraus. Daher überträgt sich die Forderung zur Einbeziehung wiederverwendungsunterstützender Konzepte in gleicher Weise auf den Entwurf der zugrundeliegenden mathematischen Modelle. Generische Konzepte sollen zur Anwendung kommen, um ausgehend von einer Analysephase zunächst geeignete mathematische Modelle und dann in Folge die Steuerungs- und Regelungsalgorithmen zu erstellen.

In der letzten Förderungsperiode wurden als Schwerpunkte die Methodik der generischen Entwicklung objektorientierter mathematischer Modelle und die auf diesen basierende Reglerentwicklung erforscht, zur Verfügung gestellt und auf grundlegende Strukturen der Anwendungsdomäne Gebäude angewandt. Die Regelungskonzepte basieren auf parametrierbaren, komponierbaren Grundalgorithmen (PKG), bei komplexeren Anwendungen auch in mehrfach verschleifter Form. Die konsequente Weiterentwicklung der erarbeiteten Techniken und Methoden sowie der anwendungsorientierte Ausbau der zur Verfügung stehenden Modell- und Algorithmenbibliotheken ist in der zweiten Förderperiode projektiert. Zunächst ist geplant, die Algorithmenbibliothek durch Aufnahme höherer Regelungsalgorithmen wie Mehrgrößenregler, Entkopplungsregler, prädiktiver Regler, Regler auf Basis von Fuzzy Logik [3], [6] auszubauen. Bislang werden solche Konzepte im Bereich der Haustechnik kaum eingesetzt, obwohl sie gerade hier, aufgrund der Vielzahl zu beeinflussender, verkoppelter Prozesse ideal geeignet wären. Wiederverwendungsunterstützende Konzepte werden bisher bei höheren Algorithmen kaum berücksichtigt.

Das Teilprojekt setzt sich hier zwei Ziele. Zunächst ist zu detektieren, wann der Einsatz höherer Regelungsalgorithmen Vorteile gegenüber dem bisherigen Konzept mehrfach verschleifter, einfacher Grundalgorithmen aufweist bzw. wann er notwendig ist. Eine Vermeidung ihres Einsatzes spart Komplexität und unterstützt indirekt die Wiederverwendbarkeit durch alleinigen Einsatz der modularen Grundalgorithmen. Mittels Simulationsmodellen ist daher in typischen, exemplarischen Konfigurationen die Erfordernis höherer Regelungsstrategien zu überprüfen. Die in der Simulation bewährten höheren Algorithmen werden in die Algorithmenbibliothek übernommen. Die Untersuchungen werden zugleich dazu dienen, geeignete Attribute zur Selektion des optimalen Algorithmus aufgrund hervorstechender Streckencharakteristika zu entwickeln und die einzulagernden Algorithmen damit auszustatten. Bei ähnlichen Anwendungen wird die Nutzung gleicher Regelungskonzepte als geeignet erscheinen. Formale Ähnlichkeitskriterien auf der Anforderungsseite sowie Eignungskriterien auf der Algorithmenseite sind zu entwickeln und zu spezifizieren.

Eine Herausforderung wird die Einbeziehung wiederverwendungsunterstützender, generischer Entwurfskonzepte zur Erstellung der höheren Regelungsstrategien für konkrete Anwendungsfälle sein. Die Methode soll ebenfalls auf der Selektion, Adaption und Komposition modularer, in einer Bibliothek abgelegter Bausteine beruhen. Geeignete, wiederverwendungsorientierte Beschreibungsformen sind zu entwickeln. Die Aufgabe ist weitaus komplexer als im Falle einfacher komponierbarer Grundalgorithmen. Dies rührt daher, dass modellgestützte Regler, wie z.B. die Entkopplungsregler das konkrete mathematische Streckenmodell innerhalb des eigentlichen Algorithmus beinhalten. Die Adaption allein über einstellbare Parameter wird nicht möglich sein.

Beruhte die Selektion bisher auf der Qualität der zu erzielenden Regelung, so soll künftig auch der Erstellungsaufwand betrachtet werden. Es sind geeignete Metriken zu definieren, um die Rentabilität des generischen Entwurfes der diversen höheren Regelungsalgorithmen abschätzen und vorhersagen zu können. Diese Prognose sollte auch als Kriterium zweier Entscheidungen dienen. Erstens, ist die Anwendung eines höheren Algorithmus gegenüber einem Algorithmus auf Basis parametrierbarer, komponierbarer Grundalgorithmen zu präferieren? Und zweitens, ist im Falle der Entscheidung für den Einsatz eines höheren Algorithmus, dieser als Unikat neu zu entwerfen, oder ist er mittels systematischer Anwendung eines wiederverwendungsunterstützenden Entwurfsverfahrens zu entwickeln?

b) *Teilprojektübergreifende Unterstützung als Infrastrukturprojekt*

Auch in der nächsten Förderperiode wird davon ausgegangen, dass ein fester Prozentsatz von D2 im Sinne einer infrastrukturellen Unterstützung eingesetzt wird (siehe *Ergebnisbericht D2*). Hierbei geht es um Expertenwissen aus den Bereichen Heizung, Klima, Lüftung sowie der Steuerungs- und Regelungstechnik. Teilprojekt D2 übernimmt hierbei die Rolle des „Anwendungsexperten“ und bietet als Serviceprojekt den anderen Teilprojekten projektübergreifende Unterstützung an. Das Expertenwissen wird den anderen Projekten in geeignet aufbereiteter Form zur Verfügung gestellt.

Fallstudien und Experimente bekommen so eine praxisorientierte, realitätsnahe Komponente. Zudem soll von D2 Domänenwissen in geeigneter Form in der Erfahrungsdatenbank des SFBs abgelegt werden. Die spezielle domänenspezifische Form dieser allgemein nutzbaren Einträge sowie die Problematik der Selektion gehören zu den Forschungsgegenständen des Teilprojektes D2.

3.3 Stand der Forschung

Eine detaillierte Zusammenstellung findet sich im Projektantrag der letzten Förderungsperiode. Neue Aspekte ergaben sich im Bereich mathematischer Modellierung zur Simulation technischer Prozesse. Gerade zur Beschreibung großer, komplexer Systeme werden zunehmend objektorientierte mathematische Ansätze entwickelt ([2], [9] - [12], [16]). Ihre modulare Struktur, klare Strukturierung (Vererbung, Aggregation, Schnittstellendefinitionen, Komponentenkapselung) sowie die nicht berechnungskausale Implementierung [16] machen sie zu idealen Werkzeugen, um Wiederverwendung und generische Erzeugung in die Entwurfsmethodik mathematischer Modelle zur Beschreibung technischer Prozesse zu integrieren. Sie haben in den letzten drei Jahren Eingang in die Forschungslandschaft gefunden und hier bereits z.T. die älteren Werkzeuge wie Matlab/Simulink ersetzt.

Kaum verändert gegenüber der letzten Förderperiode hat sich der Stand bei der Anwendung höherer Regelungsalgorithmen. Diese sind seit Jahren Forschungsgegenstand, sowohl aus methodischer als auch aus applikativer Sicht [1], [5]. In den Vordergrund des Forschungsinteresses sind dabei insbesondere modellprädiktive Regler gerückt [1]. Generell interessiert bei den höheren Algorithmen die Entwurfsmethodik derartiger Regler, verbunden mit ihrer Anwendbarkeit auf praktische Probleme [5]. Die Beurteilung des Nutzens geschieht nach der Performance des Regelungssystems, d.h. nach den – meist simulativ nachgewiesenen - Regelungsergebnissen. Der Aspekt der Wiederverwendbarkeit war, abgesehen von der Entwurfsmethodik, bisher von minderem Interesse. Nur punktuell – z.B. bei Regelung von Destillationskolonnen [7] – findet auch Code-Wiederverwendung statt. Dies verwundert zum einen, da der recht hohe Entwicklungsaufwand derartiger Regler durch Wiederverwendung kompensiert werden könnte. Andererseits ist es auch verständlich, da der Wiederverwendungsaspekt den ohnehin komplizierten Entwurfsprozess zunächst nicht einfacher macht. Es ist dem SFB 501 zuzuschreiben, die Denkweise des Reuse in systematischer Weise nun auch auf diesen Teil reaktiver Software auszudehnen.

Literatur

[1] Allan et al.: Crude unit control and optimisation at Grangemouth Refinery. In: Computing & Control Engineering Journal (10) 5, 1999.

[2] Beater P.: Objektorientierte Modellierung physikalischer Systeme Teil 10. at1, A1-A5 (2000)

[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 predicitve 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 predicitve 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 Verkopplungen 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.

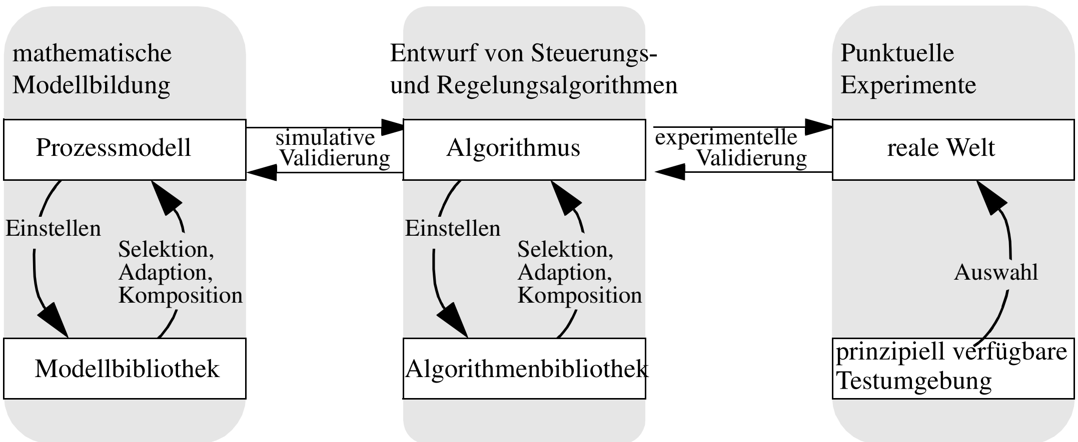

Abbildung 1: Veranschaulichung des modellbasierten Reglerentwurfes

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.

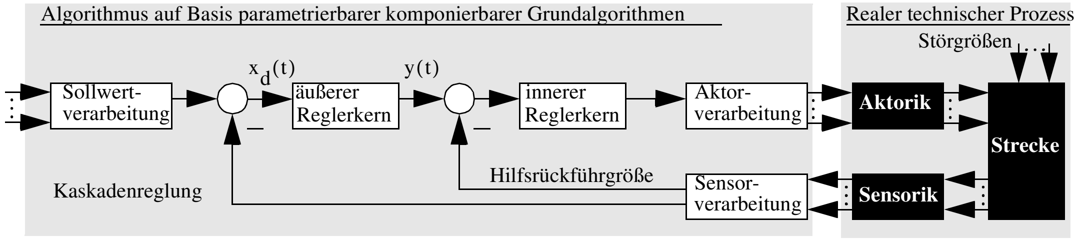

Abbildung 2: Prinzipskizze eines verschleiften Reglers aus parametrierbaren, komponierbaren Grundalgorithmen (PKG)

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, reuseunterstützenden 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 instantiierter 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 Komponentenmodelle 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, instantiierter 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.

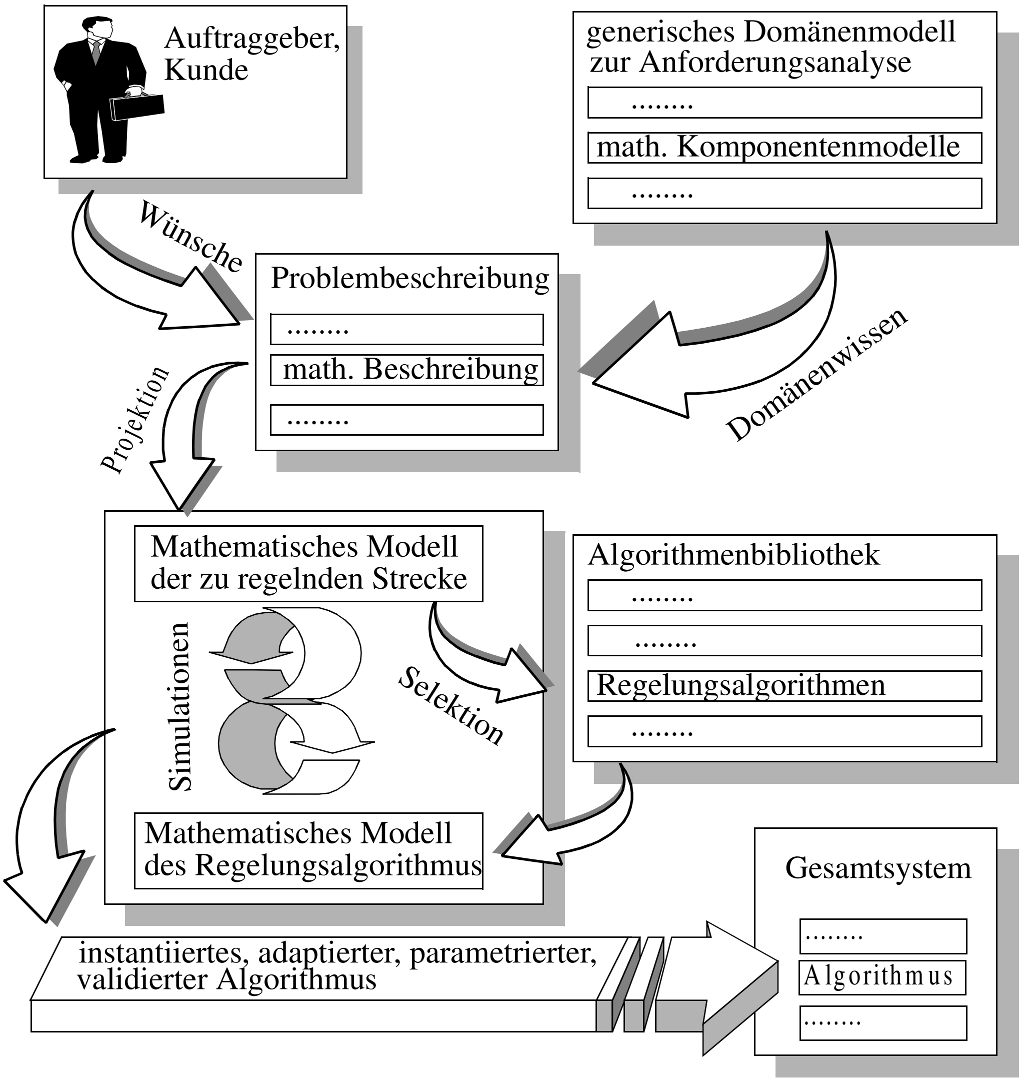

Abbildung 3: Vorgehensweise zur generische Algorithmenentwicklung

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 heraus projizieren. 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 wissensspeicherndere 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.

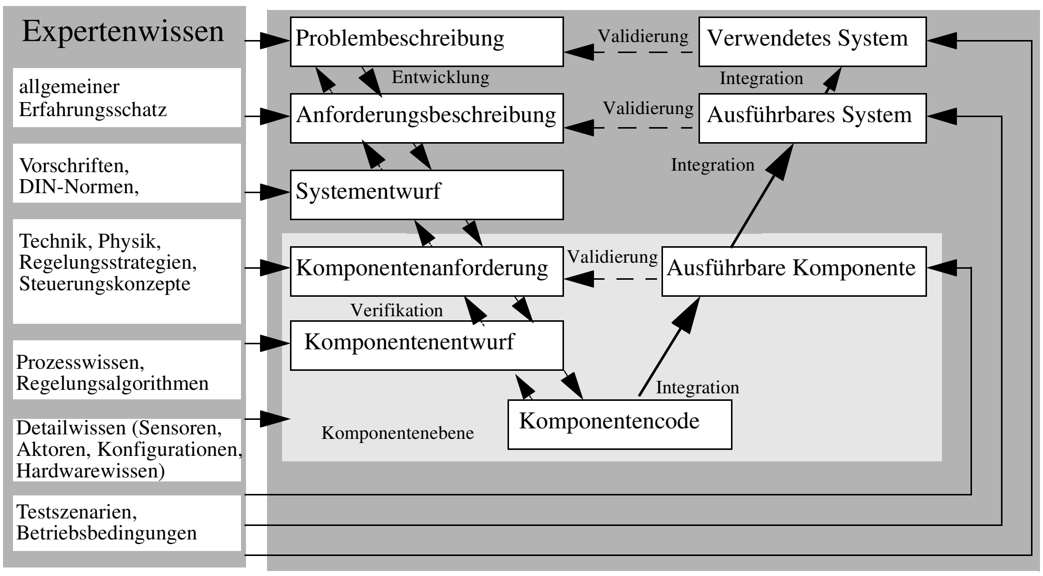

Abbildung 4: Integration von Expertenwissen in allen Phasen des Softwareentwicklungsprozesses, hier exemplarisch dargestellt am Beispiel des V-Modells.

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 gegebenfalls 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 genutzer Standardkomponenten. Modelle diverser thermischer Komponenten sowie der Fluiddynamik 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 Adaptionsaufwand 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 abgeleitetet 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

<table>
<tr><th>Personal</th><th>2001
1. H.J.</th><th>2001
2. H.J.</th><th>2002
1. H.J.</th><th>2002
2. H.J.</th><th>2003
1. H.J.</th><th>2003
2. H.J.</th></tr>
<tr><td>Grundausstattung</td><td colspan="6"></td></tr>
<tr><td>A. Diehl, wiss. Mit.</td><td colspan="6">A1,A2, A3 und A4</td></tr>
<tr><td></td><td colspan="3"></td><td colspan="2">B4</td><td></td></tr>
<tr><td></td><td colspan="5"></td><td>D3</td></tr>
<tr><td>P. Denu, wiss. Mit.</td><td colspan="2">B1</td><td></td><td colspan="2">B4</td><td></td></tr>
<tr><td></td><td colspan="2"></td><td>C2</td><td colspan="2"></td><td>D2</td></tr>
<tr><td></td><td colspan="5"></td><td>D3</td></tr>
<tr><td>H. Voos, wiss. Mit.</td><td colspan="2">B2</td><td></td><td colspan="2">B4</td><td></td></tr>
<tr><td></td><td colspan="2"></td><td>C2</td><td colspan="2"></td><td>D2</td></tr>
<tr><td></td><td colspan="5"></td><td>D3</td></tr>
<tr><td>Ergänzungsausstattung</td><td colspan="6"></td></tr>
<tr><td>Dr. R. Merz, wiss. Mit.</td><td colspan="6">A1,A2, A3 und A4</td></tr>
<tr><td></td><td colspan="3"></td><td colspan="2">B4</td><td></td></tr>
<tr><td></td><td colspan="5"></td><td>D3</td></tr>
<tr><td>NN., wiss. Mit.</td><td colspan="2">B1, B2, B3</td><td></td><td colspan="2">B4</td><td></td></tr>
<tr><td></td><td colspan="2"></td><td>C1</td><td colspan="2"></td><td>D1</td></tr>
<tr><td></td><td colspan="5"></td><td>D3</td></tr>
<tr><td>NN., stud. Hilfskraft</td><td colspan="6">A1,A2, A3 und A4</td></tr>
<tr><td></td><td colspan="5"></td><td>D3</td></tr>
<tr><td>NN., stud. Hilfskraft</td><td>A1</td><td colspan="5"></td></tr>
<tr><td></td><td></td><td colspan="5">B1, B2, B3 und B4</td></tr>
<tr><td></td><td colspan="5"></td><td>D3</td></tr>
</table>

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 Grundausstattung

Eine explizite Aufgabenzuordnung der Mitarbeiter der Grundausstattung 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 Ergänzungsausstattung

Eine explizite Aufgabenzuordnung der Mitarbeiter der Ergänzungsausstattung 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 Ergänzungsausstattung 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/Ausgabeblöcken 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 *Control Design Tools*	6 000.-	4 000.-	4 000.-
1 *kommerzielle Gebäudesimulatoren*	3 000.-	3 000.-	3 000.-
1 *Realtime Workshop*	2 000.-		
1 *Mess- und Steuerungs-PC*		4 000.-	
1 *PC Target 1.0-Softwarepaket*	2 000.-		
1 *DA/AD-Wandlerkarten*	2 000.-	3 000.-	
1 *Aktorik und Sensorik*	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	-	-	-