

Projektbereich B

**Generische Modelle und  
Methoden der Systementwicklung**

### **3 Projektbereich B: Generische Modelle und Methoden der Systementwicklung**

#### **3.01 Zusammenfassende Darstellung des Projektbereiches**

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

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

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

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

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

**3.02 Bisherige und beantragte Förderung des Projektbereiches**

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

(jeweils Ergänzungsausstattung, Beträge in DM 1000,-)



### 3.1 Allgemeine Angaben zum Teilprojekt B1

#### 3.1.1 Thema

Generische Modellierung von Prozessen und Experimenten

#### 3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Software Engineering

#### 3.1.3 Leiter<sup>1</sup>

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

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

#### 3.1.4 -entfällt-

#### 3.1.5 In dem Teilprojekt sind vorgesehen

|                                        |                                 |                                              |
|----------------------------------------|---------------------------------|----------------------------------------------|
| Untersuchungen am Menschen             | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| klinische Studien im Bereich           |                                 |                                              |
| der somatischen Zell- oder Gentherapie | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| Tierversuche                           | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| gentechnologische Untersuchungen       | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |

---

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

### 3.1.6 Bisherige und beantragte Förderung des Teilprojektes

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

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

## 3.2 Zusammenfassung

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

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

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

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



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

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

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

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

### 3.3 Stand der Forschung

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

#### Literatur

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



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

### 3.4 Eigene Vorarbeiten

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

## Literatur

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

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

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

### **3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan**

#### **3.5.1 Ziele und Methoden**

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

##### **Generierung umfassender Projektpläne**

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

Folgende Hauptprobleme sind jedoch mit einer solchen Vorgehensweise verbunden:

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

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

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

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

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

Im Kontext dieses Forschungsschwerpunkts existieren folgende Forschungsfragen:

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

### **Generierung von Experimentfamilien**

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

Beim Einsatz der experimentellen Methodik wurden folgende Hauptprobleme identifiziert:

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

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

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

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

In diesem Zusammenhang existieren folgende Forschungsfragen:

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

### **Prozessleitstand**

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

Hierbei wurden folgende Hauptprobleme identifiziert:

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



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

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

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

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

In diesem Forschungsschwerpunkt existieren folgende Forschungsfragen:

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

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

### 3.5.2 Arbeitsprogramm

#### **Arbeitsschwerpunkt A: Generierung umfassender Projektpläne**

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

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

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

Personalbedarf: 18 PM

##### **Arbeitspaket A2:** Integration weiterer Plansichten

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

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

Personalbedarf: 12 PM

### **Arbeitsschwerpunkt B: Generierung von Experimentfamilien**

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

#### **Arbeitspaket B1:** Analyse relevanter Einflussfaktoren von Experimentplänen und Experimentiertechniken

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

Personalbedarf: 6 PM

#### **Arbeitspaket B2:** Generische Auslegung Experimentplänen

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

Personalbedarf: 6 PM

### **Arbeitspaket B3: Generierung von Experimentvarianten**

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

Personalbedarf: 10 PM

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

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

Personalbedarf: 9 PM

### **Arbeitsschwerpunkt C: Prozessleitstand**

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

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

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

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

Personalbedarf: 18 PM

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

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

Personalbedarf: 21 PM

**Arbeitsschwerpunkt D: Evaluierung entwickelter Techniken und Werkzeuge**

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

**Arbeitspaket D1:** Evaluierung der Methodik zur Generierung von umfassenden Projektplänen

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

Personalbedarf: 6 PM

**Arbeitspaket D2:** Evaluierung der Methodik zur Generierung von Experimentfamilien

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

Personalbedarf: 6 PM

**Arbeitspaket D3:** Evaluierung des Prozessleitstands

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

Personalbedarf: 6 PM

### 3.5.3 Zeitplan

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

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

### 3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

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

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

### 3.7 Ergänzungsausstattung für das Teilprojekt

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

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

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

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



## 3.7.1 Begründung des Personalbedarfs

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

### **Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung**

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

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

### **Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung**

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

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

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

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

### 3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

|                                                                                                                                                                 | 2001                                                                                                                                                                                    | 2002  | 2003  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| Für sächliche Verwaltungsausgaben stehen als <b>Grundausstattung</b> 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 <b>Ergänzungsausstattung</b> beantragt<br>(entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7) | 3.000                                                                                                                                                                                   | 3.000 | 3.000 |

#### Begründung zur Ergänzungsausstattung der Sächlichen Verwaltungsausgaben

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

### 3.7.3 Investitionen (Geräte über 20.000 brutto)

| Bezeichnung des Gerätes<br>(ggf. mit Typenbezeichnung) | Einzelpreis<br>2001 | Einzelpreis<br>2002 | Einzelpreis<br>2003 |
|--------------------------------------------------------|---------------------|---------------------|---------------------|
|                                                        | -                   | -                   | -                   |
| Summe                                                  | -                   | -                   | -                   |



### 3.1 Allgemeine Angaben zum Teilprojekt B2

#### 3.1.1 Thema

Flexible Planung und Steuerung von Softwareentwicklungsprozessen

#### 3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Wissensbasierte Systeme und Künstliche Intelligenz

#### 3.1.3 Leiter

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

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

#### 3.1.4 -entfällt-

#### 3.1.5 In dem Teilprojekt sind vorgesehen

|                                        |                                 |                                              |
|----------------------------------------|---------------------------------|----------------------------------------------|
| Untersuchungen am Menschen             | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| klinische Studien im Bereich           |                                 |                                              |
| der somatischen Zell- oder Gentherapie | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| Tierversuche                           | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| gentechnologische Untersuchungen       | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |

### 3.1.6 Bisherige und beantragte Förderung des Teilprojektes

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

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

## 3.2 Zusammenfassung

Langfristiges Ziel des Teilprojekts B2 ist die Entwicklung von Methoden und Techniken zur flexiblen Planung und Steuerung komplexer Software-Entwicklungsprozesse. Hierbei liegt der Hauptaugenmerk darauf, Aktivitäten, die sich mit dem Management des Softwareentwicklungsprozesses beschäftigen (also Meta-Aktivitäten sind), möglichst fließend in die Projektdurchführung zu integrieren. In den ersten beiden Förderungsperioden wurden hierzu Techniken zur flexiblen Abwicklungsunterstützung und zur Unterstützung von Planverfeinerungen und Planänderungen während der Abwicklung realisiert. In Zusammenarbeit mit dem Teilprojekt B1 wurde die Software-Prozess-Modellierungssprache MILOS<sup>1</sup> entwickelt und auf deren Basis in der zweiten Förderungsperiode in Zusammenarbeit mit dem Teilprojekt A2 ein gleichnamiges Workflow-Management-System implementiert. Das MILOS-System erlaubt die Modellierung und Planung von Software-Entwicklungsprojekten und unterstützt ihre Durchführung u.a. mittels To-Do-Listen und Änderungsnotifikationen. Planverfeinerungen und Planänderungen während der Projektdurchführung werden unterstützt und gehen in die dem MILOS-System zugrundeliegende Abhängigkeitsverwaltung ein.

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

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

1. Modeling Language and Operational Support for Software Processes

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

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

### 3.3 Stand der Forschung

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

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

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

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

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

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

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

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

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

### **3.4 Eigene Vorarbeiten**

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



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

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

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

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

### 3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

#### 3.5.1 Ziele und Methoden

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

#### Meta-Modellierung

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

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

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

### **Erstellung und Wiederverwendung von Plan- und Ausführungstraces**

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

### **Softwareagenten zur Planungsunterstützung**

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

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

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

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

### **3.5.2 Arbeitsprogramm**

#### **Arbeitsschwerpunkt A: Unterstützung der initialen Planung**

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

#### **Arbeitspaket A1: Identifikation von Möglichkeiten zur Planverbesserung**

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

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

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

### **Arbeitspaket A2:** Generierung und Delegierung von Meta-Aufgaben

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

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

Personalbedarf: 8 PM

### **Arbeitspaket A3:** Entscheidungsabhängigkeiten und Plan Traces

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

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

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

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

Personalbedarf: 7 PM

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

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

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

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

Personalbedarf: 18 PM

**Arbeitspaket A5:** Identifikation von Möglichkeiten zur wissensbasierten Planungsunterstützung und Automatisierung

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

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

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

Personalbedarf: 10 PM

**Arbeitspaket A6:** Dynamische Teambildung

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

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

Personalbedarf: 8 PM

#### **Arbeitspaket A7: Implementierung von Softwareagenten zur Teambildung**

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

Personalbedarf: 7 PM

#### **Arbeitsschwerpunkt B: Unterstützung der Umplanung und Planverfeinerung**

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

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

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



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

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

Personalbedarf: 6 PM

### **Arbeitspaket B2:** Verzahnung von Plan- und Ausführungs-Traces

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

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

Personalbedarf: 5 PM

### **Arbeitspaket B3:** Implementierung in MILOS

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

Personalbedarf: 12 PM

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

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

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

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

Personalbedarf: 5 PM

**Arbeitspaket B5:** Verteilten Projektplanung und Ausführung in dynamischen Teams

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

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

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

Personalbedarf: 10 PM

### **Arbeitsschwerpunkt C: Praktischer Einsatz**

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

#### **Arbeitspaket C1: Einsatz im MILOS-Entwicklungsprozess**

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

Personalbedarf: 22 PM

#### **Arbeitspaket C2: Einbringung in den Gesamt-SFB**

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

Personalbedarf: 16 PM

### 3.5.3 Zeitplan

| Personal                                                                                                                                                  | 2001<br>1. H.J. | 2001<br>2. H.J. | 2002<br>1. H.J. | 2002<br>2. H.J. | 2003<br>1. H.J. | 2003<br>2. H.J. |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Grundausstattung<br><br>Kötting, Boris<br><br><br>N.N.                                                                                                    |                 |                 |                 |                 |                 |                 |
|                                                                                                                                                           | A6              |                 |                 | A7              | B5              |                 |
|                                                                                                                                                           |                 |                 |                 |                 |                 |                 |
|                                                                                                                                                           |                 |                 |                 | A7              | B5              |                 |
|                                                                                                                                                           | C1              |                 |                 | C2              |                 |                 |
|                                                                                                                                                           |                 |                 |                 |                 |                 |                 |
| Ergänzungsausstattung<br><br>Holz, Harald<br><br><br><br>Goldmann, Sigrid<br><br><br><br><br>N.N., stud. Hilfskraft<br><br><br><br>N.N., stud. Hilfskraft |                 |                 |                 |                 |                 |                 |
|                                                                                                                                                           | A3              | B2              |                 | B3              |                 |                 |
|                                                                                                                                                           |                 | A4              |                 | A5              |                 | B4              |
|                                                                                                                                                           |                 |                 |                 |                 |                 |                 |
|                                                                                                                                                           | A1              |                 | A2              |                 | A4              |                 |
|                                                                                                                                                           |                 |                 |                 | B1              |                 | B3              |
|                                                                                                                                                           |                 |                 |                 | C2              |                 |                 |
|                                                                                                                                                           | C1              |                 |                 | A7              |                 |                 |
|                                                                                                                                                           |                 | A4              |                 |                 | B3              |                 |
|                                                                                                                                                           |                 |                 |                 |                 |                 |                 |
| C1                                                                                                                                                        |                 |                 | C2              |                 |                 |                 |
|                                                                                                                                                           |                 |                 |                 |                 |                 |                 |

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

### 3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

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

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

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

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

### 3.7 Ergänzungsausstattung für das Teilprojekt

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

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

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

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

## 3.7.1 Begründung des Personalbedarfs

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

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

### **Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung**

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

### **Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung**

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

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

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



### 3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

|                                                                                                                                                                 | 2001                                                                                                                                                                                    | 2002  | 2003  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| Für sächliche Verwaltungsausgaben stehen als <b>Grundausstattung</b> 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 <b>Ergänzungsausstattung</b> beantragt<br>(entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7) | 3.000                                                                                                                                                                                   | 3.000 | 3.000 |

#### Begründung zur Ergänzungsausstattung der Sächlichen Verwaltungsausgaben

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

### 3.7.3 Investitionen (Geräte über 20.000 brutto)

| Bezeichnung des Gerätes<br>(ggf. mit Typenbezeichnung) | Einzelpreis<br>2001 | Einzelpreis<br>2002 | Einzelpreis<br>2003 |
|--------------------------------------------------------|---------------------|---------------------|---------------------|
| -                                                      | -                   | -                   | -                   |



### 3.1 Allgemeine Angaben zum Teilprojekt B4

#### 3.1.1 Thema

Generische Kommunikationssysteme

#### 3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, Rechnernetze, Datenkommunikation, Kommunikationsprotokolle, Formale Beschreibungstechniken

#### 3.1.3 Leiter

Prof. Dr. Reinhard Gotzhein

geb. 9.10.1956

Universität Kaiserslautern, Postfach 3049, 67653 Kaiserslautern

Tel.: 0631/205-3426

Fax: 0631/205-3956

Email: [gotzhein@informatik.uni-kl.de](mailto:gotzhein@informatik.uni-kl.de)

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

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

#### 3.1.4 -entfällt-

#### 3.1.5 In dem Teilprojekt sind vorgesehen

|                                        |                             |                                          |
|----------------------------------------|-----------------------------|------------------------------------------|
| Untersuchungen am Menschen             | <input type="checkbox"/> ja | <input checked="" type="checkbox"/> nein |
| klinische Studien im Bereich           |                             |                                          |
| der somatischen Zell- oder Gentherapie | <input type="checkbox"/> ja | <input checked="" type="checkbox"/> nein |
| Tierversuche                           | <input type="checkbox"/> ja | <input checked="" type="checkbox"/> nein |
| gentechnologische Untersuchungen       | <input type="checkbox"/> ja | <input checked="" type="checkbox"/> nein |

### 3.1.6 Bisherige und beantragte Förderung des Teilprojektes

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

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

## 3.2 Zusammenfassung

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

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

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

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

Darüberhinaus soll das Teilprojekt Beiträge zu den anderen Projektbereichen liefern.

### 3.3 Stand der Forschung

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

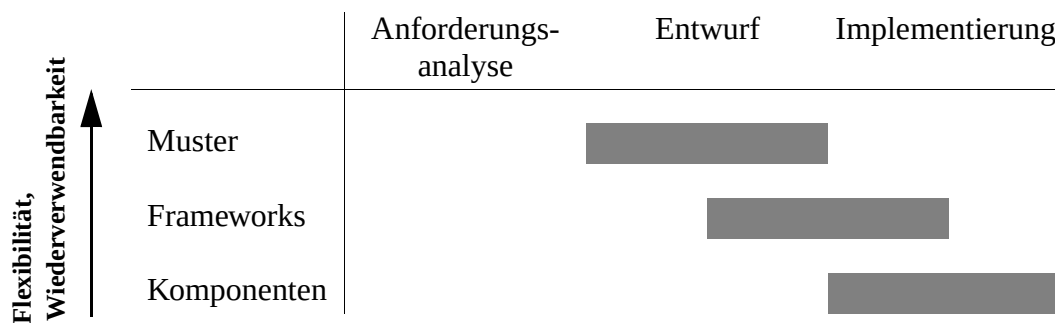
#### Komponentenbasierte Protokollentwicklung

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

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

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

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



**Abbildung 1:** Klassifikation von Wiederverwendungskonzepten

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

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

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

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

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

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

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

Im folgenden sind einige der bekannteren Ansätze zur Protokollkonfigurierung aufgeführt:

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

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

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

### **Automatische Generierung effizienter Protokollimplementierungen**

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

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

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

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



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

## Literatur

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

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

### 3.4 Eigene Vorarbeiten

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

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

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

#### Literatur

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

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

### 3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

#### 3.5.1 Ziele und Methoden

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

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

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

### **A. Musterbasierter Protokollentwurf**

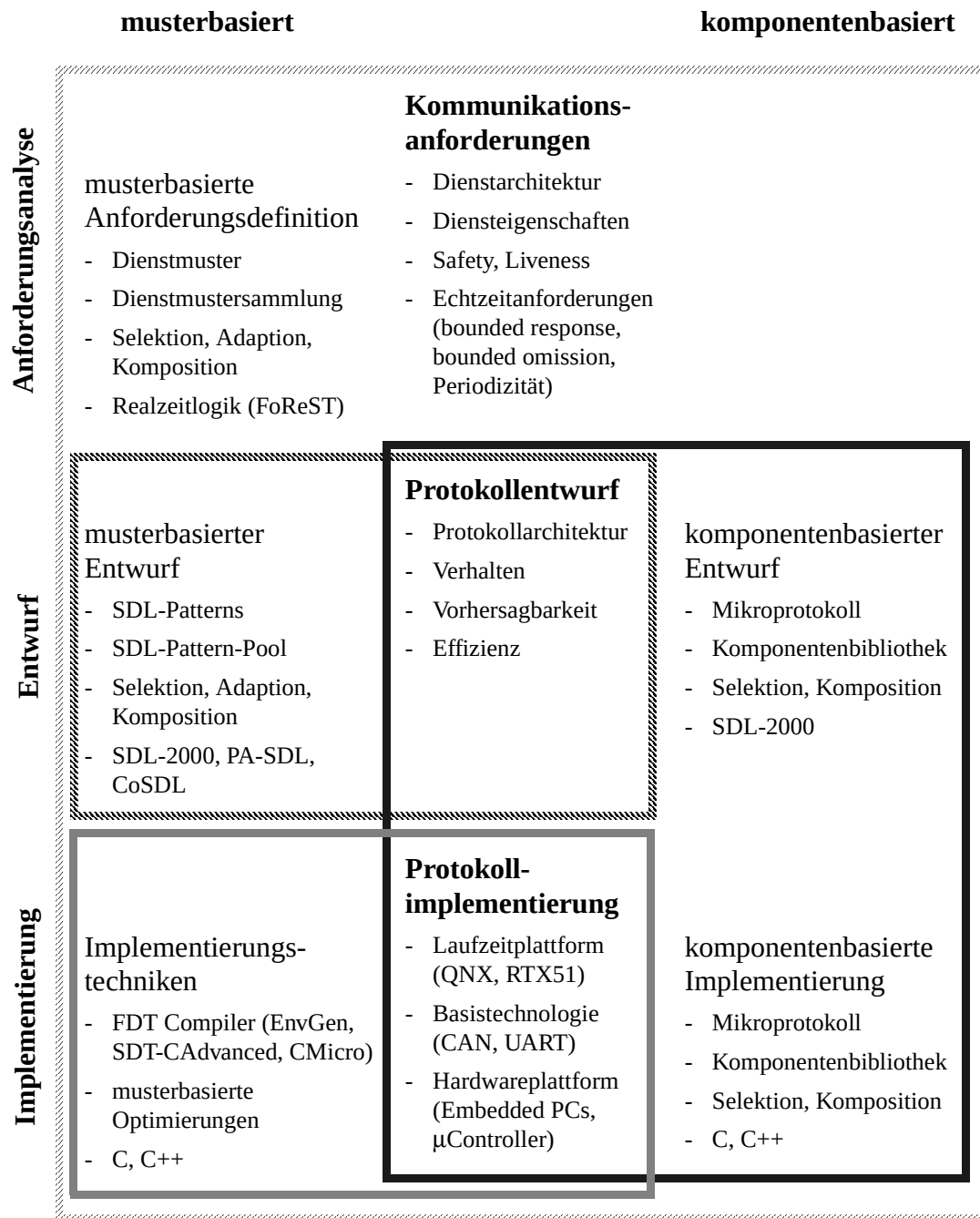
Ein wesentliches Ergebnis der vorangegangenen Förderperioden sind die Entwicklung und die praktische Erprobung des *SDL-Pattern-Ansatzes* für den Entwurf maßgeschneiderter Kommunikationsprotokolle. Bestandteile des Ansatzes sind eine Sammlung wiederverwendbarer Protokollbausteine in Form von SDL-Patterns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

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

### **B. Micro-Protocol Engineering**

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

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

Legende:

Arbeitsschwerpunkt A



Arbeitsschwerpunkt C



Arbeitsschwerpunkt B



Arbeitsschwerpunkt D

**Abbildung 2:** Aufgabenspektrum und Zuordnung zu Arbeitsschwerpunkten

sourcensituation und erforderlicher Dienstgüte unterschiedliche Mikroprotokolle komponiert werden könnten.

### **C. Ableitung effizienter Protokollimplementierungen**

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

### **D. Syntaktische und semantische Integration der Entwicklungsphasen**

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

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

### **3.5.2 Arbeitsprogramm**

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

#### **Arbeitsschwerpunkt A: Musterbasierter Protokollentwurf**

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

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

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

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

Personalbedarf: 15 PM

#### **Arbeitspaket A2:** Definition von Analysemodellen

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

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

Personalbedarf: 6 PM

#### **Arbeitspaket A3:** Pattern Mining

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



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

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

Personalbedarf: 6 PM

#### **Arbeitspaket A4:** Musterbasiertes Engineering von Mikroprotokollen

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

Personalbedarf: 18 PM

#### **Arbeitsschwerpunkt B: Micro-Protocol Engineering**

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

#### **Arbeitspaket B1:** Definition eines Beschreibungsrahmens

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

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

Personalbedarf: 3 PM

#### **Arbeitspaket B2:** Entwicklung einer Bibliothek von Mikroprotokolldefinitionen

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

Personalbedarf: 15 PM

#### **Arbeitspaket B3:** Engineering maßgeschneiderter Kommunikationsmiddleware im SFB-Anwendungsbereich

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

Personalbedarf: 9 PM

#### **Arbeitsschwerpunkt C: Ableitung effizienter Protokollimplementierungen**

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

**Arbeitspaket C1:** Untersuchung musterbasierter Implementierungstechniken

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

Personalbedarf: 9 PM

**Arbeitspaket C2:** Ableitung von Implementierungsspezifikationen

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

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

Personalbedarf: 12 PM

**Arbeitspaket C3:** Implementierung von Mikroprotokollen im SFB-Anwendungsbereich

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

Personalbedarf: 12 PM

**Arbeitsschwerpunkt D: Syntaktische und semantische Integration der Entwicklungsphasen**

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

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

#### **Arbeitspaket D1:** Integration der Entwicklungsprozesse

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

Personalbedarf: 12 PM

#### **Arbeitspaket D2:** Integration der Mustersammlungen

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

Personalbedarf: 12 PM

#### **Arbeitspaket D3:** Engineering maßgeschneiderter Kommunikationsmiddleware für skalierbare Gebäudeautomationssysteme

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

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

Personalbedarf: 15 PM

### 3.5.3 Zeitplan

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

### 3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

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

### 3.7 Ergänzungsausstattung für das Teilprojekt

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

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

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

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

## 3.7.1 Begründung des Personalbedarfs

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

### Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

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

### Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

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

### 3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

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

### Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

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

### 3.7.3 Investitionen (Geräte über 20.000 brutto)

| Bezeichnung des Gerätes<br>(ggf. mit Typenbezeichnung) | Einzelpreis<br>2001 | Einzelpreis<br>2002 | Einzelpreis<br>2003 |
|--------------------------------------------------------|---------------------|---------------------|---------------------|
|                                                        | -                   | -                   | -                   |
| Summe                                                  | -                   | -                   | -                   |



### 3.1 Allgemeine Angaben zum Teilprojekt B5

#### 3.1.1 Thema

GeneSys – Generische Systemsoftware

#### 3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Systemsoftware

#### 3.1.3 Leiter

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

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

#### 3.1.4 -entfällt-

#### 3.1.5 In dem Teilprojekt sind vorgesehen

|                                        |                                 |                                              |
|----------------------------------------|---------------------------------|----------------------------------------------|
| Untersuchungen am Menschen             | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| klinische Studien im Bereich           |                                 |                                              |
| der somatischen Zell- oder Gentherapie | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| Tierversuche                           | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| gentechnologische Untersuchungen       | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |

### 3.1.6 Bisherige und beantragte Förderung des Teilprojektes

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

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

## 3.2 Zusammenfassung

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

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

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

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

### 3.3 Stand der Forschung

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

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

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

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

### 3.4 Eigene Vorarbeiten

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

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

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

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

### Konferenzen und Workshops

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

### SFB-Berichte

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

## Diplomarbeiten

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

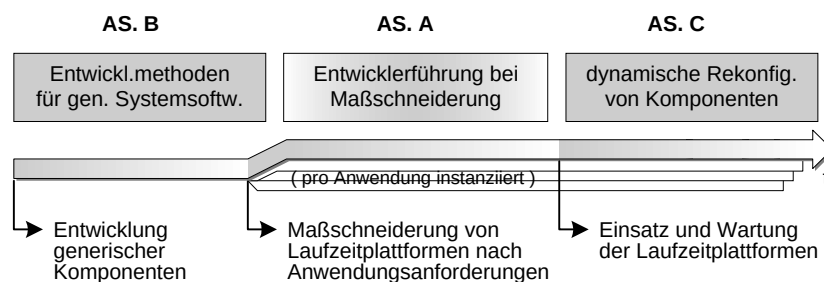
## Projektarbeiten

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

## 3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

### 3.5.1 Ziele und Methoden

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



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

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

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

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

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

### **3.5.2 Arbeitsprogramm**

#### **Arbeitsschwerpunkt A: Entwicklerführung bei der Maßschneiderung von Laufzeitplattformen**

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

##### **Arbeitspaket A1: Konfigurierbarer Gesamtprozess**

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

Personalbedarf: 12 PM



### **Arbeitspaket A2: Integrative Werkzeugunterstützung**

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

Personalbedarf: 15 PM

### **Arbeitspaket A3: Kosten-/Nutzenbewertung**

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

Personalbedarf: 12 PM

## **Arbeitsschwerpunkt B: Entwicklungsmethoden für generische Systemsoftware**

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

### **Arbeitspaket B1: Komponentenentwicklung**

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

Personalbedarf: 15 PM

### **Arbeitspaket B2: Generierung von Komponentenbeschreibungen**

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

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

Personalbedarf: 12 PM

### **Arbeitspaket B3:** Ableitung von Architekturinformationen

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

Personalbedarf: 12 PM

### **Arbeitspaket B4:** Auswahl und Evolution von Architekturen

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

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

Personalbedarf: 12 PM

### **Arbeitsschwerpunkt C: Dynamische Rekonfiguration von Komponenten**

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

#### **Arbeitspaket C1: Untersuchung von Anforderungsszenarien**

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

Personalbedarf: 12 PM

#### **Arbeitspaket C2: Erweiterung der Generik-Konzepte**

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

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

Personalbedarf: 12 PM

### **Arbeitspaket C3: Prototypische Realisierung**

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

Personalbedarf: 12 PM

### 3.5.3 Zeitplan

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

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

### 3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

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

### 3.7 Ergänzungsausstattung für das Teilprojekt

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

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

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

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

**3.7.1 Begründung des Personalbedarfs**

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



### Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

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

### Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

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

### 3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

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

### Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

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

### 3.7.3 Investitionen (Geräte über 20.000 brutto)

| Bezeichnung des Gerätes<br>(ggf. mit Typenbezeichnung) | Einzelpreis<br>2001 | Einzelpreis<br>2002 | Einzelpreis<br>2003 |
|--------------------------------------------------------|---------------------|---------------------|---------------------|
|                                                        | -                   | -                   | -                   |
| Summe                                                  | -                   | -                   | -                   |



### 3.1 Allgemeine Angaben zum Teilprojekt B10

#### 3.1.1 Thema

Anwendungsentwicklung mit vorkonfektionierten Systemen

#### 3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Systemsoftware

#### 3.1.3 Leiter

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

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

#### 3.1.4 -entfällt-

#### 3.1.5 In dem Teilprojekt sind vorgesehen

|                                        |                                 |                                              |
|----------------------------------------|---------------------------------|----------------------------------------------|
| Untersuchungen am Menschen             | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| klinische Studien im Bereich           |                                 |                                              |
| der somatischen Zell- oder Gentherapie | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| Tierversuche                           | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |
| gentechnologische Untersuchungen       | <input type="checkbox"/> ( ) ja | <input checked="" type="checkbox"/> (x) nein |

### 3.1.6 Bisherige und beantragte Förderung des Teilprojektes

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

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

## 3.2 Zusammenfassung

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

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

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

Im Rahmen des Teilprojektes werden folgende wissenschaftliche Aufgaben bearbeitet:

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

### 3.3 Stand der Forschung

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

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

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

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

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

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

### 3.4 Eigene Vorarbeiten

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

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

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

### Konferenzen und Workshops

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

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

### **Technische Berichte**

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

### **Diplomarbeiten**

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

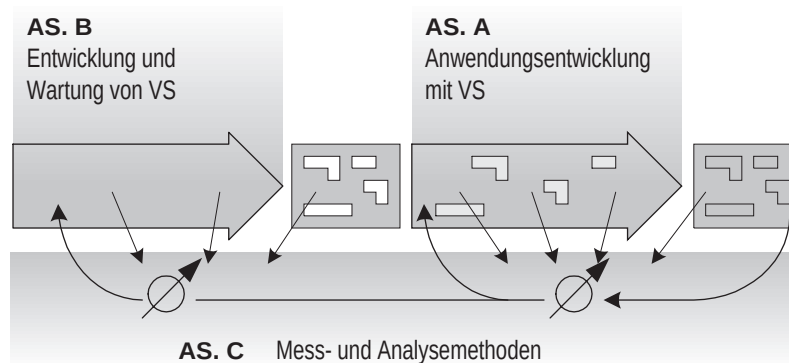
## **3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan**

### **3.5.1 Ziele und Methoden**

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

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

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



**Abbildung 1:** Übersichtsdarstellung der Arbeitsschwerpunkte

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

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

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



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

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

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

### **3.5.2 Arbeitsprogramm**

#### **Arbeitsschwerpunkt A: Anwendungsentwicklung mit vorkonfektionierten Systemen**

##### **Arbeitspaket A1: Produktmodell für Anwendungsentwicklung mit vorkonfektionierten Systemen**

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

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

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

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

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

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

Personalbedarf: 10 PM

**Arbeitspaket A2:** Prozessmodell für die Anwendungsentwicklung mit vorkonfektionierten Systemen

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

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

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

Personalbedarf: 10 PM

### **Arbeitspaket A3:** Werkzeugunterstützung für Anwendungsentwicklung mit vorkonfektionierten Systemen

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

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

Personalbedarf: 19 PM

### **Arbeitspaket A4:** Validierung der erarbeiteten Modelle und Werkzeuge

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

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

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

Personalbedarf: 15 PM

## **Arbeitsschwerpunkt B: Verfahren zur Entwicklung und Wartung vorkonfektionierter Systeme**

### **Arbeitspaket B1: Vorgehensmodell zur Entwicklung vorkonfektionierter Systeme**

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

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

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

Personalbedarf: 12 PM

### **Arbeitspaket B2:** Wartung vorkonfektionierter Systeme

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

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

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

Personalbedarf: 12 PM

### **Arbeitspaket B3:** Werkzeugunterstützung für Entwicklung und Wartung vorkonfektionierter Systeme

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

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

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

Personalbedarf: 21 PM

### **Arbeitsschwerpunkt C: Mess- und Analysetechniken**

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

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

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

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

Personalbedarf: 6 PM

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

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

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

Personalbedarf: 6 PM

### **Arbeitspaket C3:** Werkzeugunterstützung für Mess- und Analysetechniken

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

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

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

Personalbedarf: 15 PM

### 3.5.3 Zeitplan

| Personal             | 2001<br>1. H.J. | 2001<br>2. H.J. | 2002<br>1. H.J. | 2002<br>2. H.J. | 2003<br>1. H.J. | 2003<br>2. H.J. |
|----------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Grundausrüstung      |                 |                 |                 |                 |                 |                 |
|                      | C1              |                 |                 | C2              |                 |                 |
| Ergänzungsausrüstung | C3              |                 |                 |                 |                 |                 |
|                      |                 |                 |                 |                 |                 |                 |
| G. Molter            | A1 und A2       |                 |                 | A3              |                 |                 |
|                      | A4              |                 |                 |                 |                 |                 |
| L. Geyer             |                 |                 |                 |                 |                 |                 |
|                      | B1              |                 |                 | B2              |                 |                 |
| Stud. Hilfskräfte:   | B3              |                 |                 |                 |                 |                 |
|                      |                 |                 |                 |                 |                 |                 |
| N.N.                 | A3 und A4       |                 |                 |                 |                 |                 |
| N.N.                 |                 |                 |                 |                 |                 |                 |
|                      | B3 und C3       |                 |                 |                 |                 |                 |
|                      |                 |                 |                 |                 |                 |                 |



### **3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs**

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

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

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

### 3.7 Ergänzungsausstattung für das Teilprojekt

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

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

SV: Sächliche Verwaltungsausgaben (Begründung vgl. 3.7.2)

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

## 3.7.1 Begründung des Personalbedarfs

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

### Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

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

### Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

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

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

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

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

### 3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

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

### Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

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

### 3.7.3 Investitionen (Geräte über 20.000 brutto)

| Bezeichnung des Gerätes<br>(ggf. mit Typenbezeichnung) | Einzelpreis<br>2001 | Einzelpreis<br>2002 | Einzelpreis<br>2003 |
|--------------------------------------------------------|---------------------|---------------------|---------------------|
|                                                        | -                   | -                   | -                   |
| Summe                                                  | -                   | -                   | -                   |

### 3.1 Allgemeine Angaben zum Teilprojekt B11

#### 3.1.1 Thema

Zuverlässige Eingebettete Systeme

#### 3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Entwurfsmethodik Eingebetteter Systeme

#### 3.1.3 Leiter

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

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

#### 3.1.4 -entfällt-

#### 3.1.5 In dem Teilprojekt sind vorgesehen

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

#### 3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Das Teilprojekt wird neu beantragt.

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

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

### 3.2 Zusammenfassung

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

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

Man unterscheidet heute drei Ursachen für das Versagen eines eingebetteten Systems: Konstruktionsfehler, Ausfall einer Komponente und Bedienfehler. Zur Erhöhung der Zuverlässigkeit<sup>1</sup> eines solchen Systems kann man einerseits versuchen durch geeignete Maßnahmen im Entwurfsprozess (*Konstruktions-*) Fehler zu verhindern und andererseits *fehlertolerante* bzw. *robuste Systeme*, die alle drei Fehlerursachen tolerieren, entwickeln. Während andere Teilprojekte des SFB, Fortschritte bei der Fehlerverhinderung durch Qualitätssteigerung, u.a. aufgrund des SFB-spezifischen Zyklus „Messen, Bewerten, Verbessern“, erreichen wollen, soll der Schwerpunkt dieses Teilprojekts in der Entwicklung fehlertoleranter, robuster Systeme liegen.

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

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

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

Existierende Ansätze und Verfahren im Umfeld der Zuverlässigkeitsbetrachtung sol-

---

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

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

### **3.3 Stand der Forschung**

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

#### **Entwurfsmethodik eingebetteter Systeme**

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

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

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

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

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

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

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

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



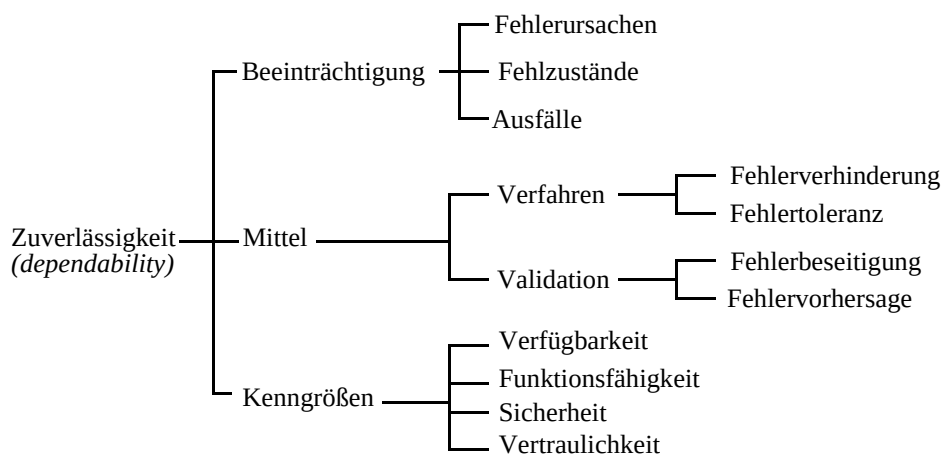
weise arbeitet auch die Forschungsgruppe von Prof. Nett aus Magdeburg gemeinsam mit der GMD und anderen Partnern an einer integrierten Entwurfsumgebung für Echtzeitsysteme [4]. Als weiteres Beispiel sei das in München in der Forschungsgruppe von Prof. Färber entwickelte REAR-Framework genannt, bei dem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

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

### Zuverlässigkeit und Robustheit eingebetteter Systeme

Sicherheit, Zuverlässigkeit und Robustheit<sup>1</sup> sind wesentliche Merkmale eingebetteter Systeme. Sie betreffen sowohl das technische System selbst - einschließlich der mechanischen und elektronischen Komponenten - als auch die eingebettete Software, die in der Regel das technische System überwacht und steuert, wobei der Mensch als Bediener nicht unberücksichtigt bleiben darf. Vorarbeiten auf dem Gebiet des Entwurfs zuverlässiger Systeme findet man daher zum einen in den Ingenieurfächern, die sich mit den technischen Systemen beschäftigen, und zum anderen im Umfeld der Automatisierungstechnik und des Software-Engineering. Auch wenn die Betrachtung der Zuverlässigkeit stark praxisorientiert ist und die verschiedenen Ingenieurdisziplinen ihre eigenen Begriffe und Lösungen verwenden, gibt es bereits einen großen

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



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

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

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

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

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

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

### **Fehlerverhinderung, Qualitätssicherung**

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

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

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

### **Fehlertoleranz, Robustheit**

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

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

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

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

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

Ein heutzutage weit verbreiteter Ansatz ist FME(C)A (Failure Modes Effects and Criticality Analysis [8]). Es handelt sich hierbei um ein Werkzeug zur Identifikation, Analyse und Priorisierung potentieller Fehler. Ausgehend von sehr frühen Arbeiten Ende der 50er Jahre ist FME(C)A heute integraler Bestandteil aller QS 9000<sup>1</sup>-Qualitätssysteme. Man findet diesen Ansatz bereits in einer Reihe kommerzieller Systeme.

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

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

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

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

---

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

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

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

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

## Literatur

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

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

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

- [34] J. G. Silva, L. M. Silva, H. Madeira, J. Bernardino, *A Fault-Tolerant Mechanism for Simple Controller*, Proc. First European Dependable Computing Conference, Springer Verlag, Berlin, 1991
- [35] Sonderforschungsbereich 358 „Automatisierter Systementwurf“, <http://www.iee.et.tu-dresden.de/~sfb358/>
- [36] VDI, *Sicherheitstechnik und Automatisierung*, VDI-Bericht 1336, VDI-Verlag, Düsseldorf, 1997

### 3.4 Eigene Vorarbeiten

Erfahrungen und Vorarbeiten auf dem beantragten Gebiet wurden vor allem in Projekten des SFB 124 „VLSI-Entwurfsmethoden und Parallelität“ und im Laufe der ersten beiden Förderungsperioden dieses SFB erlangt bzw. durchgeführt.

Im Sonderforschungsbereich 124 hatten wir uns im Teilprojekt B4 “Chip Planning” unter anderem mit ECAD<sup>1</sup>-Entwicklungsumgebungen beschäftigt. Obwohl wir uns dabei vor allem mit dem Entwurf hochintegrierter Schaltungen beschäftigten, findet man doch viele Parallelen zum Entwurf eingebetteter Systeme, dem Umfeld dieses Förderantrags. Unserer Meinung nach lassen sich die damals gewonnenen grundlegenden Überlegungen im Bereich der Entwurfstheorie [37], [43] - auf denen letztendlich die ECAD-Entwurfsumgebung PLAYOUT [45] basiert - auch auf den Entwurf eingebetteter Systeme verallgemeinern bzw. erweitern.

PLAYOUT [45] ist eine Entwurfsumgebung, bei der eine Vielzahl unterschiedlicher externer und selbstgeschriebener Entwurfswerkzeuge über eine zentrale Datenverwaltungs- und Entwurfsmanagement-Komponente gekoppelt sind. Ähnlich zum projektspezifischen Teil der im aktuellen SFB entwickelten Erfahrungsdatenbank werden in PLAYOUT alle Entwurfsdaten zentral verwaltet, wobei die Schnittstelle zur Datenverwaltungskomponente allen Entwurfswerkzeugen ihre eigene Sicht auf den Datenbestand ermöglicht.

Ein für diesen Antrag sehr wichtiger Aspekt bildeten die Arbeiten in Bezug auf ein interaktives Entwurfsmanagement. Basierend auf einer Analyse des aktuellen Entwurfszustandes und einer Abschätzung des weiteren Entwurfsverlaufs wird eine Planung der folgenden Entwurfsschritte ausgeführt und ein oder mehrere Entwerfer mit den nächsten Entwurfsaufgaben beauftragt. Die letztendlichen Entwurfsentscheidungen fällt allerdings der Mensch, so dass das PLAYOUT-Entwurfsmanagement eine Entscheidungsunterstützung darstellt und keinen vollautomatischen Entwurf anstrebt.

Die dem Entwurf zugrunde liegende Entwurfstheorie basiert auf dem Prinzip der schrittweisen Verfeinerung in einer zweidimensionalen Entwurfshierarchie [44]. Sie ist Grundlage der PLAYOUT-Produkt- und -Prozessmodellierung und wird zur Zeit

---

1. ECAD = Electronic CAD



im aktuellen SFB im Teilprojekt D1 und in projektübergreifenden Entwicklungsteams (Team 3.1 und 3.2) auch im Bereich des Entwurfs von Steuerungssystemen exemplarisch untersucht. Langfristiges Ziel dieses Teilprojektes ist es u.a., gemeinsam mit anderen Teilprojekten, ein Produkt- und Prozessmodell für alle Phasen des Entwurfs zuverlässiger eingebetteter Systeme basierend auf dem Prinzip der schrittweisen Verfeinerung zu entwickeln.

Praxisnahe Arbeiten im Bereich robuster, zuverlässiger Systeme wurden bei der Entwicklung zweier anwendungsspezifischer Parallelrechner durchgeführt. Gefördert im SFB 124 wurde im Teilprojekt B4 sowohl die Schaltung eines Parallelrechners zur effizienten Ausführung von Produktionssystemen [42] als auch ein weitgehend lauffähiger Prototyp eines Parallelrechners zur schnellen Ausführung von Verfahren aus dem Bereich der Bildverarbeitung (Echtzeit-Volumenvisualisierung) [39] entwickelt. Bei der ersten Rechnerarchitektur wurden in den frühen Phasen des Entwurfs Systemkomponenten, die in Bezug auf die Zuverlässigkeit des Systems kritisch werden könnten, identifiziert. Durch Einführung von Redundanz konnte anschließend die Fehlertoleranz erhöht werden. Bei der Entwicklung des Parallelrechners für die Bildverarbeitung wurden ebenfalls Untersuchungen durchgeführt, inwieweit die Verfügbarkeit der Architektur durch Redundanzbildung erhöht werden kann, die Ergebnisse flossen allerdings aus Aufwandsgründen nicht in den gebauten Prototyp ein.

Die Erfahrungen aus dem ECAD-Bereich sollen in dieses Teilprojekt einfließen. Dies betrifft neben dem allgemeinen Entwurfsansatz der schrittweisen Verfeinerung vor allem den Aspekt der Systempartitionierung und das Mapping von Systemfunktionalitäten auf Komponenten des verteilten Zielsystems. Hier möchten wir versuchen, die (Re-) Partitionierung von VLSI-Systemkomponenten [45] auf die Partitionierungsphase des Hardware-/Software Codesign auszudehnen. Ein Großteil der zur Partitionierung notwendigen Verfahren und Kostenfunktionen sind in den beiden Bereichen sehr ähnlich [40]. Die (nicht-funktionalen) Zuverlässigkeitsaspekte der zu entwerfenden Systeme betreffen in beiden Anwendungsbereichen einerseits den gesamten Entwicklungszyklus, werden andererseits schwerpunktmäßig in der Design-Phase in funktionale Anforderungen und Systemeigenschaften überführt.

Im aktuellen SFB wurden in den ersten beiden Förderungsperioden wesentliche Vorarbeiten im Teilprojekt D1, das vom Antragssteller mitgeleitet wurde, und über die Mitarbeit von D1 im Querschnittsprojekt Q1 geleistet. Hier wurde die Zuverlässigkeit exemplarisch vor allem in der Analysephase untersucht.

Während der ersten beiden Förderungsperioden wurden bei den projektübergreifenden Beispielenentwürfen die Aspekte Sicherheit und Zuverlässigkeit explizit in der Anforderungsanalysephase mit modelliert und berücksichtigt [38], [41]. Diese expliziten Betrachtungen der nicht-funktionalen Zuverlässigkeitsanforderungen waren wichtige Vorarbeiten in Bezug auf die schrittweise Überführung der Anforderungen in entsprechende Entwurfsentscheidungen in einem möglichst späten Entwicklungsschritt. Aus den Erfahrungen der Teamentwürfe lassen sich Hinweise für die Bestim-

mung geeigneter Entwicklungsphasen für einzelne Zuverlässigkeitsbetrachtungen ableiten.

## Literatur

- [37] J. Altmeyer, B. Schürmann, *Design Formalization and Retrieval of Reuse Candidates*, J.S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design '96*, 231-250, Kluwer Academic Publishers, 1996
- [38] R. L. Feldmann, J. Münch, S. Queins, S. Vorwieger, G. Zimmermann, *Baselining a Domain-Specific Software Development Process*, SFB 501-Bericht Nr. 02/99, Universität Kaiserslautern, 1999
- [39] J. Lichtermann, *Eine Architektur zur Echtzeitvisualisierung von Volumendaten*, Dissertation, Fachbereich Informatik, Universität Kaiserslautern, 1998
- [40] S. Queins, T. Tetteroo, B. Schürmann, *Bewertung des dynamischen Verhaltens von SDL-Modellen*, SFB 501-Bericht Nr. 09/99, Universität Kaiserslautern, 1999
- [41] S. Queins, G. Zimmermann, *A First Iteration of a Reuse-Driven, Domain-Specific System Requirements Analysis Process*, SFB 501-Bericht Nr. 13/99, Universität Kaiserslautern, 1999
- [42] F. Schreiner, *PESA-I - A parallel distributed-memory architecture for production systems*, Dissertation, Fachbereich Informatik, Universität Kaiserslautern, 1992
- [43] B. Schürmann, *Modeling Design Data and Design Processes in the PLAYOUT CAD Framework*, Current Issues in Electronic Modeling (CIEM), Band 6, Kluwer Academic Publishers, Juni 1996
- [44] B. Schürmann, J. Altmeyer, M. Schütze, *On Modeling Top-Down VLSI Design*, Proc. IEEE Int. Conference on Computer Aided Design (ICCAD), San Jose, California, 1994
- [45] B. Schürmann, G. Zimmermann. *PLAYOUT - A Hierarchical Layout System*, in G. Zobrist (ed.) *VLSI Design Environments*, Gordon and Breach Science Publishers, S. 1-52, Amsterdam, 2000

## 3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

### 3.5.1 Ziele und Methoden

Eingebettete Systeme werden als korrekt erachtet, wenn sie ihrer Spezifikation genügen. Dies nachzuweisen ist eines der Anliegen vieler Teilaktivitäten im SFB. Hierbei wird die Annahme gemacht, dass sich die Umgebung der eingebetteten Software, d.h. die Hardware, die Benutzer und die Umwelt, wohl verhält. Wenn diese Annahme

jedoch wegfällt, womit man bei eingebetteten Systemen immer rechnen muss, so sollte das System nach Möglichkeit in der Lage sein, mit dem unerwarteten Fehlverhalten zurecht zu kommen. Man spricht hierbei von Robustheit und erhöhter Zuverlässigkeit.

Diese Thematik des Entwurfs robuster, zuverlässiger Systeme soll Thema dieses neu beantragten Teilprojekts sein. Hiermit werden zwei Aspekte, die bisher im SFB noch nicht ausreichend betrachtet wurden, angegangen. Zum einen soll anhand eines konkreten Effekts untersucht werden, wie nicht-funktionale Anforderungen systematisch in den Systementwicklungsprozess aufgenommen werden können. Dabei ist unser Ziel, durch die Trennung der funktionalen Anforderungen von den nicht-funktionalen (in diesem Fall speziell der Zuverlässigkeit) zu generischen, wieder-verwendbaren Prozessen und Modellen zu gelangen, die die zukünftige Entwicklung zuverlässiger eingebetteter Systeme wesentlich vereinfachen. Ein zweites umfassendes Ziel des Teilprojekts liegt in der Betrachtung der Zuverlässigkeit an sich. Sie ist unseres Erachtens nicht nur ein beliebiges Beispiel für nicht-funktionale Systemeigenschaften, sondern nimmt im Bereich der eingebetteten Systeme, wofür sich der SFB interessiert, neben den Zeitanforderungen eine herausragende Stellung ein. Bei der Entwicklung geeigneter Anwendungsentwicklungsprozesse sollte daher die Zuverlässigkeit nicht außer Acht gelassen werden.

Langfristiges Ziel des Teilprojekts ist die Betrachtung von Sicherheit und Zuverlässigkeit eingebetteter Systeme während des gesamten Entwurfprozesses. Sicherheit und Zuverlässigkeit haben Auswirkungen auf alle Phasen der Anwendungsentwicklung und bilden einen Schwerpunkt in der Design-Phase. In diesem Teilprojekt soll untersucht werden, wie bestehende Ansätze zur Entwicklung und Modellierung robuster, zuverlässiger und sicherer Systeme in den Systementwicklungsprozess einfließen können. Die Untersuchung formaler Aspekte der Design-Phase soll in enger Zusammenarbeit mit Teilprojekt C1 durchgeführt werden.

In den Mittelpunkt unserer Betrachtungen sollen die Bewertung der Verfügbarkeit und Funktionsfähigkeit entworfenen Systeme sowie die Erhöhung der Zuverlässigkeit durch Redundanz und Notlaufeigenschaften gestellt werden. Dabei sollen nicht (notwendigerweise) neue Bewertungsverfahren und Fehlertoleranzansätze entwickelt werden. Vielmehr wollen wir existierende Ansätze und Verfahren im Umfeld der Zuverlässigkeitsbetrachtung schrittweise in den Entwicklungsprozess des SFB einfließen lassen und so eine stufenweise Erhöhung der Zuverlässigkeit der im SFB entwickelten Systeme erreichen.

Aus den genannten Fragestellungen ergeben sich drei Aufgaben und Ziele, die im Folgenden beschrieben sind und deren Umsetzung in zwei Arbeitsschwerpunkten im Abschnitt 3.5.2 „Arbeitsprogramm“ erläutert wird. Unsere Ziele sind

- Verbesserung der Fehlertoleranz und Notlaufeigenschaften eingebetteter Systeme,
- Bewertung der Zuverlässigkeit,
- Untersuchung der Design-Phase.

Bei der Umsetzung dieser Ziele wird ein erster Arbeitsschwerpunkt die Erweiterung

des gesamten Systementwicklungsprozesses um die explizite Betrachtung von Zuverlässigkeitsanforderungen umfassen. Den zweiten Arbeitsschwerpunkt wird die eingehende Untersuchung der Design-Phase bilden. Hier haben die Sicherheits- und Zuverlässigkeitsanforderungen starken Einfluss auf die Systemstruktur, Komponentenauswahl und Redundanz.

### **Fehlertoleranz und Notlaufeigenschaften eingebetteter Systeme**

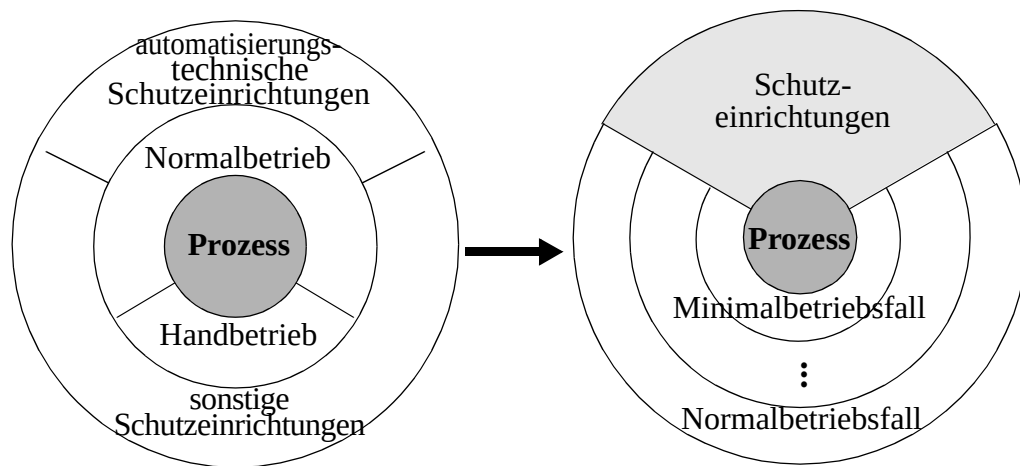
Grundlage unserer Überlegungen sind die drei Ursachen für ein Systemversagen, d.h., dass die Komponenten eines eingebetteten Systems fehlerhaft entworfen sein können, dass Hardwarekomponenten (mit einer mehr oder weniger großen Wahrscheinlichkeit) ausfallen können und dass mit Bedienfehlern zu rechnen ist. Diesen Problemen muss bei der Forderung nach einem robusten und zuverlässigen System Rechnung getragen werden.

Der in der Regel nicht ganz zu erfüllenden Forderung nach einem ausfallfreien System wird durch zwei Verfahren begegnet. In Subsystemen, die aus Sicherheits- oder anderen Gründen nicht ausfallen dürfen, soll durch Einführung von Fehlererkennungsmaßnahmen und Redundanz die Fehlertoleranz erhöht werden. Um Kosten zu sparen, soll dies jedoch nur für ausgewählte, wichtige Teilsysteme erfolgen. Für das Gesamtsystem interessiert uns vielmehr die Frage, inwieweit die Robustheit und Fehlertoleranz des Systems durch die Einführung von Notlaufeigenschaften („stepwise degradation“, „fail-safe“ bzw. „fail-operational states“) erhöht werden können. Gedacht ist an eine Hierarchie von Betriebsfällen bezüglich des Verhaltens der verschiedenen Systemfunktionalitäten, beginnend mit einem „Normalbetriebsfall“, der das Idealverhalten des Systems charakterisiert, und endend mit einem „Minimalbetriebsfall“, der das absolute Minimum von Eigenschaften beschreibt (vgl. Abbildung 1). Für alle Betriebsfälle bzw. -modi müssen entsprechende Zuverlässigkeitsanforderungen spezifiziert und in geeignete funktionale Systemeigenschaften überführt werden. Ob Entwurfsentscheidungen alle Zuverlässigkeitsanforderungen erfüllen, muss durch geeignete Bewertungsverfahren nachgewiesen werden.

### **Bewertung der Zuverlässigkeit**

Zur Bewertung der Zuverlässigkeit eines (teil-) entworfenen Systems müssen dessen Eigenschaften mit den Zuverlässigkeitsanforderungen verglichen werden. Hierbei sind einerseits stochastische Aussagen bzgl. der Ausfallwahrscheinlichkeit von Komponenten und Systemfunktionalitäten zu treffen und andererseits Auswirkungen von Fehlern auf das Systemverhalten zu untersuchen.

Die zuverlässigkeitsbedingten Systemanforderungen sollen für einzelne Aufgaben Angaben darüber enthalten, mit welcher Wahrscheinlichkeit eine Aufgabe erfüllt werden muss (→ Verfügbarkeitsanforderung) und eventuell darüber hinaus, durch welche (Notlauf-) Aufgabe sie im Fehlerfall ersetzt werden soll (→ Fail-Operational-Anforderung). Die erlaubte Ausfallwahrscheinlichkeit einer Aufgabe des Minimalbetriebsfalls liegt nahe bei Null und wird in Richtung Normalbetriebsfall ansteigen.



**Abbildung 1:** Einführung einer Hierarchie von Betriebsfällen  
Links ist das in [30] beschriebene Schalenmodell zur Abgrenzung von Automatisierungsmaßnahmen dargestellt. Die beiden Betriebsfälle „Normalbetrieb“ und „Handbetrieb“ sollen durch eine Hierarchie von Betriebsfällen ersetzt werden.

Zur Auswertung der Verfügbarkeit werden physikalische Systemeigenschaften benötigt. Diese werden im Laufe der Systementwicklung in immer umfangreicherem Maße bekannt (der Großteil der Entscheidungen, die zu diesen physikalischen Eigenschaften führen, wird in der Design-Phase getroffen, weswegen diese Phase besonders betrachtet werden soll). Die Entwurfsqualität (→ Teilprojekt A1) und die bekannten physikalischen Systemeigenschaften sollen in ein geeignetes Fehlermodell einfließen, das den Entwurfsentscheidungen zugrunde gelegt bzw. mit dessen Hilfe die geforderte Verfügbarkeit überprüft werden kann. Das Fehlermodell wird entwurfsbegleitend verwendet, immer dann, wenn Entscheidungen über die physikalische Systemstruktur getroffen werden. Damit lassen sich Verletzungen bezüglich der Verfügbarkeit frühzeitig erkennen und Entscheidungen gegebenenfalls revidieren.

Zur Bewertung der Auswirkungen von Fehlern auf das Gesamtsystem sind Fehlerfortpflanzungsmodelle zu verwenden. Mit ihrer Hilfe hoffen wir, das Gesamtsystem so partitionieren zu können, dass sich Fehler möglichst nur lokal auswirken. Dies wird eine wesentliche Aufgabe der Design-Phase sein.

Ein wesentliches Ziel im Rahmen der Bewertung und Erhöhung der Zuverlässigkeit ist das Herausarbeiten generischer, wiederverwendbarer Aspekte. Langfristig sollen wiederverwendbare Modelle und Prozesse entwickelt werden, mit deren Hilfe sich die Zuverlässigkeit neu entwickelter eingebetteter Systeme verbessern lässt.

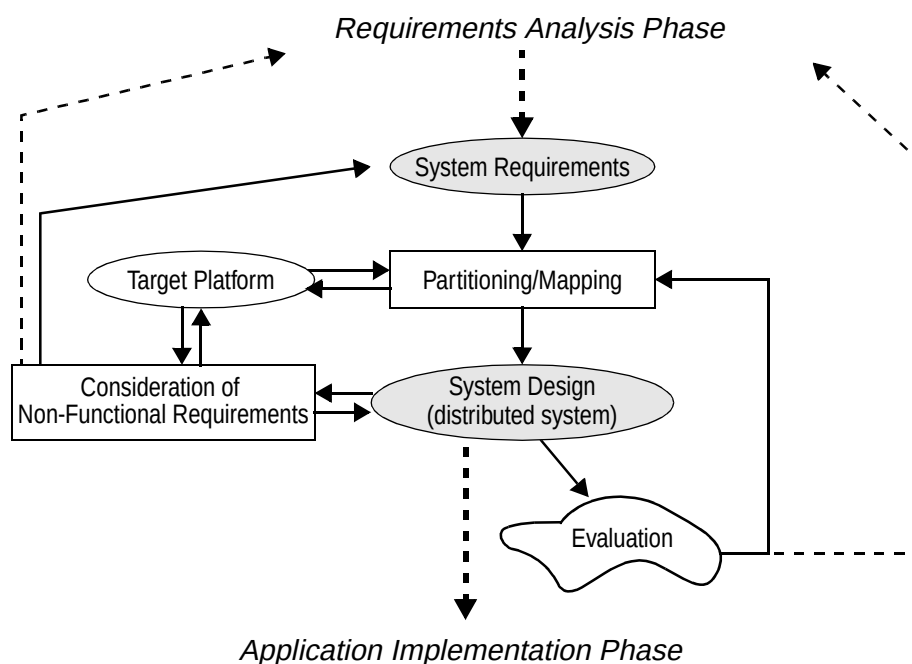
### Design-Phase

In der zweiten Förderungsperiode wurde im SFB die Anforderungsanalysephase verstärkt untersucht. Mit den nun vorliegenden Ergebnissen soll in der dritten Förderungsperiode auch die Design-Phase intensiver betrachtet werden. Die Design-Phase ist für dieses Teilprojekt besonders wichtig, da in dieser Phase die nicht-funktionalen

Aspekte eine große Rolle spielen. Sie sind ausschlaggebend für die Systemstruktur (vor allem der hardwarenahen Schichten).

In der Design-Phase (Abbildung 2) ist das zu entwerfende System geeignet in Hardware- und Softwarekomponenten zu partitionieren. Diese Aufgabe fällt in das weite Feld des Hardware/Software-Codesign. Allerdings sind die Fragestellungen in unserem Fall von anderer Natur als bei den meisten anderen Untersuchungen. Es geht uns weniger darum, zeitkritische Aufgaben in Hardware zu bearbeiten, als vielmehr eine geeignete Laufzeitplattform für die Softwarekomponenten des Anwendungssystems zu finden. Die Zielarchitektur wird ein stark verteiltes System aus Controllern, Sensoren und Aktuatoren sein.

Die Fehlertoleranz ist in der Design-Phase durch Einfügen von Redundanz und Notlaufmaßnahmen den Anforderungen anzupassen. Hierzu sollen bekannte Verfahren zur Fehlererkennung, zum Einfügen von Redundanz bzw. Notlaufmaßnahmen und zur Bewertung der Zuverlässigkeit an die Architekturen der betrachteten Domäne



**Abbildung 2:** Vereinfachtes Prozessmodell der Design-Phase.

In einem iterativen Verfeinerungsprozess werden die (funktionalen) Systemanforderungen auf eine Zielplattform (Hardware und Systemsoftware) abgebildet. Nicht-funktionale Anforderungen werden schrittweise in strukturelle und funktionale Anforderungen/Eigenschaften umgesetzt, was die Systemanforderung und die Zielplattform verändert/erweitert. Die Bewertung eines Entwurfsschritts kann zu Iterationen in der Design-Phase, aber auch zu Rückschritten zu früheren Entwicklungsphasen führen.

Parallel zur Design- und Implementierungsphase der Anwendung erfolgt Design und Implementierung der Hardware sowie der Betriebs- und Kommunikationssysteme (Zielplattform).

angepasst werden. Langfristiges Ziel ist auch hier die Suche nach passenden generischen, wiederverwendbaren Modellen und Prozessen.

Insgesamt sollen folgende Fragen in diesem Teilprojekt beantwortet werden:

- Explizite Trennung funktionaler und nicht-funktionaler Anforderungen in den frühen Phasen des Systementwurfs und schrittweise Ersetzung der nicht-funktionalen Anforderungen in funktionale Anforderungen bzw. funktionale Systemeigenschaften.
- Geeignete Modellierung der Zuverlässigkeitsanforderungen und Notlaufeigenschaften in allen Systementwicklungsphasen. Festlegung, welche Entscheidungen im Hinblick auf die nicht-funktionalen Anforderungen wann getroffen werden müssen. Untersuchung der Wechselwirkungen zwischen dem Anwendungsentwurf einerseits und Systemstruktur, Hardware sowie Kommunikations-/Betriebssysteme andererseits. Daraus ergibt sich eine Erweiterung der Produkt- und Prozessmodelle sowie der Entwurfstheorie.
- Untersuchung der Überprüfbarkeit von Zuverlässigkeitsanforderungen mit Hilfe bekannter Fehlermodelle. Übernahme bekannter Maßnahmen zur Fehlererkennung und zum Einfügen von Redundanz in Hardware- und Softwarekomponenten zur Erfüllung der Zuverlässigkeitsanforderungen.
- Bearbeitung von Notlaufeigenschaften.
- Erarbeitung generischer, wiederverwendbarer Prozesse und Modelle.
- Sicherstellung der Verfolgbarkeit von Zuverlässigkeitsanforderungen.
- Einbettung des Ansatzes in den SFB-Entwicklungsprozess. Dies bietet die Möglichkeit der Werkzeugunterstützung - u.a. Test- und Validierungswerkzeuge. Darüber hinaus Einbindung der Zuverlässigkeitsbetrachtung in einen umfassenden Prototyping-Ansatz.

### 3.5.2 Arbeitsprogramm

Zur Umsetzung der oben genannten Ziele werden zwei Arbeitsschwerpunkte definiert. Im Arbeitsschwerpunkt A soll der bisherige SFB-Systementwicklungsprozess um den Aspekt der Zuverlässigkeit erweitert werden. Da die nicht-funktionalen Anforderungen im Wesentlichen die Design-Phase betreffen, wird diese im Arbeitsschwerpunkt B gesondert betrachtet. In beiden Arbeitsschwerpunkten sollen die Zuverlässigkeitsanforderungen zu robusten, fehlertoleranten Systemen führen, wobei die nicht-funktionalen Anforderungen im Laufe des Systementwicklungsprozesses schrittweise in funktionale Anforderungen bzw. Eigenschaften des Systems überführt werden.

## **Arbeitsschwerpunkt A: Erweiterter Systementwicklungsprozess**

In diesem Arbeitsschwerpunkt soll der Entwurf zuverlässiger eingebetteter Systeme über alle Entwurfsphasen hinweg betrachtet werden. Hierzu werden zwei Aspekte vordergründig verfolgt, die in den beiden Arbeitspaketen A1 und A2 bearbeitet werden. Dies sind zum einen die Aufbereitung der Problembeschreibung zur Herausarbeitung der Zuverlässigkeitsaspekte, die in allen Entwurfsphasen schrittweise in funktionale Eigenschaften überführt werden können. Der zweite Aspekt ist die Einbindung der (um Zuverlässigkeitsaspekte erweiterten) Design-Phase in den gesamten Systementwicklungsprozess. Auch wenn der größte Teil der Zuverlässigkeitsaspekte die Design-Phase betreffen, soll in diesem Arbeitsschwerpunkt untersucht werden, welche Zuverlässigkeitsanforderungen in anderen Entwicklungsphasen bearbeitet werden sollten. Die Arbeiten haben starke Beziehungen zu der Entwicklung einer (Entwurfs-) Systematik, die im Teilprojekt D1 zusammen mit C1 ausgeführt werden soll. In einem dritten Arbeitspaket soll gemeinsam mit Teilprojekt D1 das Prototyping im Zusammenhang mit Zuverlässigkeit untersucht werden.

### **Arbeitspaket A1: Zuverlässigkeits- und Notlaufanforderungen**

Schwerpunktmäßig sollen in diesem Arbeitsschwerpunkt die frühen Entwicklungsphasen betrachtet werden. Der Hauptaugenmerk lag im SFB bisher bei der Betrachtung der funktionalen Anforderungen während der Analysephase. Sicherheits- und Zuverlässigkeitsaspekte wurden exemplarisch mit den funktionalen Aspekten gleichbehandelt. In diesem Arbeitspaket sollen nun die funktionalen und nicht-funktionalen Aspekte voneinander getrennt werden, wobei im Bereich der Zuverlässigkeit eine hierarchische Modellierung von Notlaufeigenschaften („graceful degradation“) erarbeitet werden soll (vgl. Ziel „Fehlertoleranz und Notlaufeigenschaften“). Es soll untersucht werden, inwieweit der dem SFB zugrunde liegende objektorientierte Entwicklungsansatz es ermöglicht, in sich abgeschlossene Module zu konstruieren, deren Spezifikationen auch alle Fehlermöglichkeiten enthalten. Dazu soll die funktionale Spezifikation um notwendige Operationen für den Fehlerfall erweitert werden.

Personalbedarf: 8 PM

### **Arbeitspaket A2: Erweiterung der Entwurfsprozesse um Zuverlässigkeit**

In diesem Arbeitspaket sollen die Ergebnisse des Arbeitsschwerpunkts B („Design-Phase“) auf den gesamten Entwicklungsprozess erweitert werden. Zum einen soll das Prozessmodell der Design-Phase in den Gesamtprozess eingebunden werden, zum anderen soll untersucht werden, welche Zuverlässigkeitsanforderungen in welchen Entwicklungsschritten genauer betrachtet werden sollten. Dieses Arbeitspaket kann erst in der zweiten Hälfte der Förderungsperiode, wenn erste Ergebnisse aus dem Arbeitsschwerpunkt B vorliegen, begonnen werden.

Ein langfristiges Ziel, das in diesem Arbeitspaket gelöst werden soll, ist in diesem Zusammenhang die Extraktion generischer, wiederverwendbarer Prozesse und Modelle aus durchgeführten Arbeiten anhand exemplarischer Entwürfe.

Personalbedarf: 6 PM



**Arbeitspaket A3: Prototyping**

Dieses Arbeitspaket ist eng verknüpft mit Arbeitspaket A2, da auch hier der gesamte Entwicklungsprozess im Vordergrund steht. Darüber hinaus liegt hier eine enge Kooperation mit Teilprojekt D1 vor, in dem unter anderem ein durchgängiger, entwerfsbegleitender Prototyping-Ansatz untersucht wird. Dieses Prototyping ist insbesondere solange wichtig, wie es noch nicht möglich ist, die gesamte Systementwicklung zu verifizieren.

Der Prototyping-Ansatz des SFB soll um das Phänomen Zuverlässigkeit erweitert werden. Dadurch erhält Prototyping eine neue Qualität. Durch die Systemsimulation anhand eines Prototypen können in Zusammenarbeit mit dem Kunden fehlende Notlaufforderungen festgelegt werden.

Personalbedarf: 4 PM

**Arbeitsschwerpunkt B: Design-Phase**

Der größte Teil aller Zuverlässigkeitsanforderungen wird während der Design-Phase in funktionale und strukturelle Systemeigenschaften umgesetzt. Daher bildet die Design-Phase einen eigenen Arbeitsschwerpunkt.

Zur Umsetzung der Zuverlässigkeitsanforderungen müssen diese mit den aktuellen Systemeigenschaften verglichen werden, wofür entsprechende Fehlermodelle ausgewertet werden müssen (Arbeitspaket B1). Ausgehend von der Systemanforderungsspezifikation und der Bewertung einer angenommenen Systemstruktur mit Hilfe der genannten Fehlermodelle muss eine modifizierte Spezifikation für die (restliche) Design-Phase hergeleitet werden (Arbeitspaket B2). Diese führt nach weiteren Entwurfsschritten (Partitionierung, Abbildung der Funktionalitäten auf Systemkomponenten) zu einem Entwurfsergebnis, das wieder bewertet werden muss. Die zuletzt genannten Schritte sollen in diesem Teilprojekt exemplarisch in Experimenten durchgeführt werden (Arbeitspaket B3).

**Arbeitspaket B1: Bewertung der Zuverlässigkeit**

Der Kern der Design-Phase bei der Entwicklung eingebetteter Systeme ist aufgrund der Vielzahl unterschiedlicher Optimierungskriterien ein iterativer Verbesserungsprozess. Nach jedem durchgeführten Design-Schritt muss das System bewertet und gegebenenfalls nach bestimmten Kriterien verbessert werden. Zur Bewertung der Zuverlässigkeit eines Entwurfs sollen existierende Modelle zur Anwendung kommen. Diese müssen ausgewählt und an die Produktmodelle des SFB angepasst werden. Hierbei müssen die Fehlermodelle, die in der Regel nur zwischen Betriebs- und Ausfallzuständen unterscheiden, auf die geplante Hierarchie von Notlaufeigenschaften angepasst werden.

Erarbeitete Ergebnisse dieses Arbeitspakets müssen anschließend auf den gesamten Entwicklungsprozess übertragen werden (vgl. Arbeitspaket A2).

Personalbedarf: 8 PM

**Arbeitspaket B2:** Umsetzung der Zuverlässigkeits- und Notlaufanforderungen  
in strukturelle und funktionale Anforderungen

Die bis zum Beginn der Design-Phase nicht bearbeiteten Zuverlässigkeits- und Notlaufanforderungen werden in zwei Schritten in ein System-Design überführt. In einem ersten Schritt werden diese nicht-funktionalen Anforderungen in funktionale und strukturelle Anforderungen an das System transformiert. Diese Anforderungen betreffen einerseits die Fehlererkennung, andererseits Redundanz- und Notlaufaspekte. Das Ergebnis ist eine modifizierte Systemanforderungsspezifikation, auf der dann weitere Entwurfsentscheidungen aufsetzen. Einige der nicht-funktionalen Anforderungen sind an den Betriebssystem- und Kommunikationssystementwurf weiterzureichen bzw. in Kooperation mit diesen zu betrachten (→ Teilprojekte B4 und B5).

Personalbedarf: 6 PM

**Arbeitspaket B3:** Entwurfsumgebung und Design-Experimente

Nachdem alle nicht-funktionalen (Zuverlässigkeits- und Notlauf-) Anforderungen in funktionale und strukturelle Anforderungen überführt wurden (Arbeitspaket B2), ist eine geeignete Systemstruktur zu finden. Zentrale Operationen sind dabei die Partitionierung der (funktionalen) Objekte, die Festlegung einer geeigneten Hardware-Umgebung (soweit diese noch nicht festliegt) und die Abbildung der Funktionalitäten auf die Systemkomponenten. Diese Aufgaben fallen in das große Gebiet des Hardware/Software-Codesign und sollen in diesem Arbeitspaket exemplarisch in gezielten Experimenten durchgeführt werden. Das Ziel ist hierbei ein durchgängiger Prozess, der die Zuverlässigkeitsaspekte in geeigneter Weise berücksichtigt. Zur Durchführung der Experimente wird eine interaktive Entwurfsumgebung benötigt, die in diesem Arbeitspaket prototypisch entwickelt werden soll.

Gegen Ende der Förderungsperiode soll der erreichte Stand der Forschung durch ein größeres, teilprojektübergreifendes Experiment nachgewiesen werden.

Personalbedarf: 40 PM

### 3.5.3 Zeitplan

| Personal                             | 2001<br>1. H.J.         | 2001<br>2. H.J.     | 2002<br>1. H.J. | 2002<br>2. H.J. | 2003<br>1. H.J. | 2003<br>2. H.J. |  |
|--------------------------------------|-------------------------|---------------------|-----------------|-----------------|-----------------|-----------------|--|
| Grundausrüstung<br>Torsten Tettersoo |                         |                     |                 |                 |                 |                 |  |
|                                      |                         |                     |                 |                 | AP A2           |                 |  |
|                                      | AP B1                   |                     |                 |                 |                 |                 |  |
|                                      | AP B2                   |                     |                 |                 |                 |                 |  |
|                                      | AP B3 - Implementierung |                     |                 |                 |                 |                 |  |
|                                      | AP B3 - Experimente     |                     |                 |                 |                 |                 |  |
| Ergänzungsausrüstung<br>N.N.         |                         |                     |                 |                 |                 |                 |  |
|                                      | AP A1                   |                     |                 |                 |                 |                 |  |
|                                      |                         |                     |                 |                 | AP A2           |                 |  |
|                                      |                         |                     | AP A3           |                 |                 |                 |  |
|                                      | AP B3 - Implementierung |                     |                 |                 |                 |                 |  |
|                                      |                         | AP B3 - Experimente |                 |                 |                 |                 |  |
|                                      |                         |                     |                 |                 |                 |                 |  |
| N.N.,<br>stud. Hilfskraft            | AP B3                   |                     |                 |                 |                 |                 |  |
|                                      |                         |                     |                 |                 |                 |                 |  |

### 3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

Das Teilprojekt bildet mit D1 eine Gruppe von Vorhaben, die sich die Aufgabe gestellt hat, verbesserte Systementwicklungsprozesse aus der gewählten Anwendungsdomäne zu entwickeln. Während D1 einen Entwicklungsansatz basierend auf Prototyping mit Schwerpunkt auf der Analysephase verfolgt, sollen in diesem Teilprojekt geeignete Verfahren zur Steigerung der Zuverlässigkeit der Anwendungssysteme in die Produkt- und Prozessmodelle des SFB integriert werden. Der Schwerpunkt von B11 liegt in der Design-Phase. Fragestellungen formaler Art im Rahmen der Design-Phase sollen gemeinsam mit C1 bearbeitet werden. Für Experimente soll die Testumgebung von D1 intensiv genutzt werden.

Bei der Untersuchung der Design-Phase im Allgemeinen und der Erweiterung um die nicht-funktionalen Zuverlässigkeitsanforderungen im Speziellen ist eine enge Zusammenarbeit mit B4 und B5 in Bezug auf eine geeignete Laufzeitplattform notwendig. Es ist zu untersuchen, welche Zuverlässigkeitsanforderungen durch die Anwendungssoftware zu berücksichtigen sind und welche in geeigneter Weise in Anforderungen an die Laufzeitplattform überführt werden.

Die in diesem Teilprojekt gewonnenen Verfahren und Modelle sollen langfristig in die zentrale SFB-Erfahrungsdatenbank einfließen. Dies erfordert eine Kooperation mit den Teilprojekten A1 und A3.

### 3.7 Ergänzungsausstattung für das Teilprojekt

|    | 2001                               |      |              | 2002                               |      |              | 2003                               |      |              |
|----|------------------------------------|------|--------------|------------------------------------|------|--------------|------------------------------------|------|--------------|
|    | Verg.-Gr.                          | Anz. | Betrag in DM | Verg.-Gr.                          | Anz. | Betrag in DM | Verg.-Gr.                          | Anz. | Betrag in DM |
| PK | BAT IIa                            | 1    | 104.400      | BAT IIa                            | 1    | 104.400      | BAT IIa                            | 1    | 104.400      |
|    | StdHkr.                            | 1    | 20.400       | StdHkr.                            | 1    | 20.400       | StdHkr.                            | 1    | 20.400       |
|    | zus.:                              | 2    | 124.800      | zus.:                              | 2    | 124.800      | zus.:                              | 2    | 124.800      |
| SV | Kostenkategorie oder Kennziffer    |      | Betrag in DM | Kostenkategorie oder Kennziffer    |      | Betrag in DM | Kostenkategorie oder Kennziffer    |      | Betrag in DM |
|    | (515)                              |      | -            | (515)                              |      | -            | (515)                              |      | -            |
|    | (522)                              |      | 3.000        | (522)                              |      | 3.000        | (522)                              |      | 3.000        |
|    | zusammen                           |      | 3.000        | zusammen                           |      | 3.000        | zusammen                           |      | 3.000        |
| I  | Mittel für Investitionen insgesamt |      |              | Mittel für Investitionen insgesamt |      |              | Mittel für Investitionen insgesamt |      |              |
|    | 25.000                             |      |              | -                                  |      |              | -                                  |      |              |

es bedeuten: PK: Personalbedarf und -kosten (Begründung vgl. 3.7.1)  
SV: Sächliche Veraulungsausgaben (Begründung vgl. 3.7.2)  
I: Investitionen (Geräte über DM 20.000; Begründung vgl. 3.7.3)

## 3.7.1 Begründung des Personalbedarfs

|                                                                            | Name, akad. Grad,<br>Dienststellung                              | engeres<br>Fach des<br>Mitarbeiters | Bezeichnung des Institutes<br>oder der Hochschule | Anteil der<br>aufgew.<br>Gesamt-<br>arbeitszeit<br>(bera-<br>tend: B) | im<br>SFB<br>tätig<br>seit | derzeitige<br>Einstufung<br>und<br>beantragte<br>Ver-<br>gütungs-<br>gruppe |
|----------------------------------------------------------------------------|------------------------------------------------------------------|-------------------------------------|---------------------------------------------------|-----------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------------------|
| Grundausrüstung<br>3.7.1.1<br>wiss. Mitarbeiter<br>(einschl. StdHkr.)      | (1) Schürmann, B.; Dr. habil.<br>(2) Tettersoo, T., Dipl-Inform. | Techn. Inf.<br>Techn. Inf.          | FB Inf., Uni KL<br>FB Inf., Uni KL                | 5<br>20                                                               | 1.1.95<br>1.1.95           |                                                                             |
| Ergänzungsausrüstung<br>3.7.1.3<br>wiss. Mitarbeiter<br>(einschl. StdHkr.) | (1) N.N.,<br>(2) N.N., StdHkr.                                   |                                     | FB Inf., Uni KL<br>FB Inf., Uni KL                | 38,5<br>19                                                            | 1.1.96<br>1.1.00           | Bat IIa<br>St. Hkr.                                                         |
| 3.7.1.4<br>nichtwissenschaftl.<br>Mitarbeiter                              |                                                                  |                                     |                                                   |                                                                       |                            |                                                                             |

### Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

Der Mitarbeiter der Grundausrüstung beschäftigt sich vor allem mit konzeptionellen Arbeiten im Bereich der Design-Phase und der Einbindung der Ergebnisse in den umfassenden SFB-Entwicklungsprozess. Bei der Entwicklung einer prototypischen HW/SW-Codesign-Umgebung zur Bearbeitung von Entwürfen und bei der Durchführung von Experimenten wird der Mitarbeiter vom wissenschaftlichen Mitarbeiter und von der studentischen Hilfskraft aus der Ergänzungsausrüstung unterstützt.

Von den zur Verfügung stehenden 18 PM fließen etwa zwei Drittel in die konzeptionellen Arbeiten, während ca. 6 PM für Implementierung und Experimente verfügbar sind.

### Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

Die konzeptionellen Arbeiten des Mitarbeiters der Ergänzungsausrüstung umfassen zunächst Erweiterungen des bisherigen SFB-Entwicklungsprozesses um Notlaufeigenschaften. Dies erfolgt in Zusammenarbeit mit den anderen Teilprojekten des SFB, die sich mit der Anwendungsentwicklung beschäftigen. Nachdem Ergebnisse aus dem Bereich der Design-Phase vorliegen, soll der Mitarbeiter gemeinsam mit dem Mitarbeiter der Grundausrüstung diese in den umfassenden Entwicklungsprozess integrieren. Für diese Arbeiten werden 18 PM veranschlagt.

Wesentliche Entwicklungstätigkeiten des Teilprojekts und ein Großteil der Experimente sollen vom Mitarbeiter der Ergänzungsausrüstung ausgeführt werden (18 PM), wobei er von der studentischen Hilfskraft der Ergänzungsausrüstung und vom Mitarbeiter der Grundausrüstung unterstützt wird.

### 3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

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

### Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

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

**3.7.3 Investitionen (Geräte über 20.000 brutto)**

| Bezeichnung des Gerätes<br>(ggf. mit Typenbezeichnung)                | Einzelpreis<br>2001 | Einzelpreis<br>2002 | Einzelpreis<br>2003 |
|-----------------------------------------------------------------------|---------------------|---------------------|---------------------|
| 1 Workstation HP VISUALIZE C3600<br>mit 512 MB SDRAM;<br>1 X-Terminal | 25.000              | -                   | -                   |
| Summe                                                                 | 25.000              | -                   | -                   |

Die beantragte Workstation mit dem angeschlossenen Arbeitsplatz dienen zur Versorgung des im Rahmen der Ergänzungsausstattung beantragten Mitarbeiters und der studentischen Hilfskraft.





### 3.1 Allgemeine Angaben zum Teilprojekt B12

#### 3.1.1 Thema

Stratified Component Frameworks

#### 3.1.2 Fachgebiet und Arbeitsrichtung

Informatik, AG Component Engineering

#### 3.1.3 Leiter

Prof. Dr. Colin Atkinson

geb. 14.1.1962

Universität Kaiserslautern

Postfach 3049

67653 Kaiserslautern

Tel.: 0631/205-3958

atkinson@informatik.uni-kl.de

Dr. Thomas Kühne

geb. 9.11.65

Universität Kaiserslautern

Postfach 3049

67653 Kaiserslautern

Tel.: 0631/205-4654

kühne@informatik.uni-kl.de

Die Stelle des Leiters ist befristet: ( ) nein (x) ja, bis zum 30.9.2001

#### 3.1.4 -entfällt-

#### 3.1.5 In dem Teilprojekt sind vorgesehen

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

#### 3.1.6 Bisherige und beantragte Förderung des Teilprojektes

Das Teilprojekt wird neu beantragt.

| Haushaltsjahr | Personalkosten<br>bzw. BAT-Stellen | Sächliche Verwal-<br>tungsausgaben | Investitionen | Gesamt  |
|---------------|------------------------------------|------------------------------------|---------------|---------|
| 2001          | 145.200                            | 5.000                              |               | 150.200 |
| 2002          | 145.200                            | 3.000                              |               | 148.200 |
| 2003          | 145.200                            | 3.000                              |               | 148.200 |

(jeweils Ergänzungsausstattung, Beträge in DM 1000.-)

### 3.2 Zusammenfassung

Components, frameworks, patterns and architectures have emerged as key technologies for promoting reuse in the development of software systems, and each is the subject of a great deal of ongoing research, especially within existing subprojects of the SFB 501. At present, however, it is not clear how they should be used together to achieve a maximal synergy of their respective strengths. The goal of this subproject is to investigate a practical approach for the integration of these technologies to promote reuse in the engineering of large, complex software systems. In doing so it will build upon ideas developed in previous SFB 501 subprojects, particularly those of B4, B5 and B10.

The key observation for achieving this synergy is that no single architecture (at no single level of abstraction) provides the optimal description of a system. On the contrary, different architectural presentations are optimal for different purposes. In traditional development projects, high-level (abstract) architectural descriptions produced during the refinement process are often discarded once the “real” architecture has been established. However, in framework-oriented software engineering approaches, where reuse is the prime concern, all architectural descriptions are valuable and must be carefully stored and related. We refer to a framework that contains a set of such architectural visualizations as a **stratified framework**. By using a component-oriented paradigm to construct the architectural strata in such a framework, and various kinds of patterns to establish the overall structure and relationships of the strata, an effective synergy of the aforementioned technologies can be attained. Furthermore, the tailorability and flexibility (and hence reusability) of the resulting frameworks is significantly improved. The key advantage of stratified frameworks over traditional frameworks is thus their support of reuse in terms of higher-level architectural representations of software systems.

The proposed project has four main subgoals -

- elaboration and description of the principles of framework stratification, and the properties of a stratified framework (the product),
- elaboration and description of a practical method for stratified framework development and application (the process),
- validation of the concept of framework stratification through the development of a case study (in the area of building automation),
- development of “stratification” tool support for the development and application of stratified frameworks.

In the first “research” phase, the technical and methodological foundations of the approach will be established. In the second phase, the basic premise of stratification will be validated and its key principles will be exercised. In the third phase a prototype supporting environment will be developed.

### 3.3 Stand der Forschung

Two time-honoured engineering strategies for tackling increased demand and complexity are **reuse** and **abstraction**, and over the years many techniques have been developed to apply these concepts in the context of software development. Today's leading-edge reuse technologies include -

- **components** - ready-to-run, prepackaged software capsules [1] [2]
- **frameworks** - preconfigured, semi-complete applications [3]
- **patterns** - proven solutions to common problems [4] [5]

while an abstraction technique currently attracting a great deal of attention is the concept of -

- **architecture** - a description of the fundamental structure and nature of a software system that ignores unnecessary details [6] [7]

Reuse and abstraction, in turn, are incarnations of an even more fundamental engineering principle - the **separation of concerns**. There is growing recognition in the research community that the key to improved software development practices is the strict and clean application of this principle. "Subject-Oriented Programming" [8], "Aspect-oriented development" [9], "Design Spaces" [10] and "Reflective (or meta-level) architectures" are all leading-edge development approaches motivated by the goal of highlighting and separate dimensions of concerns. Moreover, several workshops have recently sprung up with an explicit focus on separating concerns[11]. However, to date there has been little investigation of how the enhanced information and insights offered by these approaches should best be captured within frameworks to promote software reuse. The work proposed in this project aims to address this problem by investigating how to exploit the principle of separation of concerns in the development and application of component-based frameworks. In particular, it will investigate the impact of this principle on the architectural description of such frameworks.

The dominant "dimension of concern" under investigation in the project will be the "refinement" dimension, particularly the refinement of interactions. The importance of the explicit documentation of this dimension has been recognized in both formal and practical object-oriented development work. For example, Broy et al. [12] have developed a formal approach for modelling component-based systems in which the refinement of interactions plays a critical role in system structuring. Sounderajan [13], on the other hand, discusses the importance of interaction refinement in regular object-oriented software development. Stratification offers a way of systematically organizing and documenting the interaction refinements that underpin the different architectural perspectives of a system and often implicitly drive the development process.

The stratification concept also has a bearing on another important problem that is of great interest in the research community: namely, the definition of a clean model of "connectors" in the component-oriented description of system architectures [6], [3].

Most component and architecture description languages include the concept of “connectors” to capture the interaction protocols through which components, or architectural units, can be connected together. The intention is that these connectors should be “first class” citizens of the description languages that are amenable to the same set of manipulations as components themselves. The importance of supporting the first-class status of connectors results from the following observations and requirements:

- Traditional composition techniques, such as inheritance, are not suitable for dynamically “glueing” components together in a plug and play fashion
- In addition to a toolkit of readily available components, one should also possess standardized ways of combining them with a toolkit of connectors.
- Connectors can be complex, i.e., compositions of more primitive connectors.

To date, however, no entirely satisfactory model for connectors has been found. By viewing connectors as components (or collections of components) at lower architectural strata, a cleaner model for their access and manipulation may be possible. This, in turn, is expected to lead to enhanced opportunities for parameterization (i.e. “hot spot” definition). In conventional object-oriented frameworks, the “hot spots” (points of high-variation) are typically embodied as objects or classes. In other words, frameworks are typically parameterized in terms of objects within **one** architectural perspective (or strata). The object interaction mechanisms, in contrast, remain relatively fixed. However, it is well known that the functions (i.e. interactions in an object-oriented context) within software systems vary much more rapidly and significantly than objects. Indeed, this is the fundamental premise upon which the success of the object paradigm is based: objects are more stable than functions. By enabling the objects within the refinement of a high-level interaction to be points of variation (i.e. treated as “hot spots”) a stratified framework can offer more flexible parameterization than traditional object-oriented framework, and thus ultimately more flexible reuse. This will allow application engineers to influence interaction semantics, as well as object semantics, by defining specific objects tailored to the precise needs of the customer.

Since patterns will play an important role in capturing the relationships between strata, as well as the structure of the strata themselves, stratification also offers an approach for relating patterns within a “pattern language” [14]. This is regarded as the crucial step for realizing the full potential of the pattern approach. However, although architectural and design patterns are a very effective mechanism for capturing, motivating, and explaining proven design configurations, they only denote single spots in a design space and are not connected to each other very well. A pattern language, in contrast, is a generative hierarchy of patterns that narrows the choice for the application of patterns in terms of the context (i.e. the set of previously applied pattern) in which it is being applied. This recursive process is continued until the leaves of the pattern language have been reached. The patterns, and the way they are hierarchically connected, therefore guarantee the resulting design has an inherent quality. Currently, however, attempts to weave existing patterns into pattern languages are somewhat immature.

In summary, the research proposed in this thesis will make three fundamental contributions to the research focus of the SFB 501 -

- enhanced architectural abstraction mechanisms for promoting reuse in the development of large scale software systems,
- enhanced parameterization mechanisms for increasing the genericity of software frameworks, and through this, an enhanced model of “connectors”,
- enhanced pattern “languages” (or “systems”).

The resulting approach will therefore essentially offer an improved way of storing and reusing architectural knowledge in the development of software systems.

## Literatur

- [1] C. Szyperski, *Component Software - Beyond Object-Oriented Programming*, Addison-Wesley /ACM Press, 1998.
- [2] L. Baum, *Towards Generating Customized Run-time Platforms from Generic Components*, Proc. of the 11th Conf. on Advanced Systems Engineering (CAISE-99), 6th DC, 14/15 June 1999, Heidelberg, Germany.
- [3] D. D'Souza and A. C. Wills, *Catalysis: Objects, Frameworks, and Components in UML*, Addison-Wesley, 1998.
- [4] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, “*Design Patterns: Elements of Reusable Object-Oriented Software*”, Addison Wesley, 1994.
- [5] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal M., *Pattern-Oriented Software Architecture: A System of Patterns*, John Wiley and Sons Inc. 1996.
- [6] Shaw, M., and Garlan D., *Software Architecture: Perspective on an Emerging Discipline*, Prentice-Hall, 1996.
- [7] L. Baum, M. Becker, L. Geyer, G. Molter, *The Role of Architecture for Complex Systems Development*, Proc. of 11th. Int'l Conf. on Software Engineering and its Applications 1998, 8-10 Dec. 1998, Paris, France.
- [8] Harrison W. and Ossher O., *Subject-Oriented Programming (A Critique of Pure Objects)*. In *Proceedings of OOPSLA'93*, pages 411-428, ACM 1993.
- [9] Kiczales, G., J. Lampling, A. Mendhekar, C. Maeda, V. Lopez, J.-M. Loingtier and J. irwin, *Aspect-oriented Programming*, In *proceedings of the European Conference on Object-Oriented Programming*, Springer-Verlag LNCS 124, June 1997.

- [10] Baum, L., Geyer, L., Molter, G., Rothkugel, S., Sturm, P.: Architecture-Centric Software Development Based On Extended Design Spaces, Proc. 2nd ARES Workshop (Esprit IV 20477), February '98, Las Palmas
- [11] Workshop on Multi-Dimensional Separation of Concerns in Software Engineering (ICSE 2000).
- [12] Broy, M., Towards a Mathematical Concept of a Component and Its Use, TUM Report I9746, 1997.
- [13] Interaction Refinement in the Design of OO Systems, Sounderajan N., "OOPSLA'97 Workshop on Object-Oriented Behavioral Semantics", 1997.
- [14] James O. Coplien and Douglas C. Schmidt, (eds) "Pattern Languages of Program Design", Addison-Wesley, 1995.
- [15] Weiss, D. M., Lai, C. T. R. Software Product-Line Engineering. A Family-Based Software Development Process. Addison-Wesley, 1999.
- [16] Bayer, J., Flege, O., Knauber, P., Laqua, R., Muthig, D., Schmid, K., Widen, T., and DeBaud, J.-M. PuLSE: A methodology to develop software product lines. In Proceedings of the Symposium on Software Reusability (SSR'99), May 1999.
- [17] Moreira A. and S. Demeyer (Eds.), Object-oriented Technology, ECOOP'99 Workshop Reader, Lisbon Portugal, June 1999.
- [18] Feldmann R. L., Münch J., Queins S., Vorwieger S. and Zimmermann G., Baselineing a Domain-Specific Software Development Process. Technical Report No. 02/97, Sonderforschungsbereich 501, Dept. of Computer Science, University of Kaiserslautern, Germany, 1999.
- [19] Ergebnisbericht des Querschnittsprojekts Q3: "Erarbeitung eines Domänenmodells für das Anwendungsfeld Gebäudeautomation", Arbeits- und Ergebnisbericht 1998-2000, Sonderforschungsbereich 501, Fachbereich Informatik, Universität Kaiserslautern, 2000

### 3.4 Eigene Vorarbeiten

The roots of the stratification concept at the heart of the proposed project originated within the NASA MISSION project [20] [21] to help manage the complexity of large, non-stop distributed software systems such as the Space Station software. The complexity of the envisaged system called for several major distributed subsystems, including a distributed communication systems (DCS), a distributed information systems (DIS) and distributed policy management system (DPS). Architectural stratification proved to be the most effective way for visualizing the role and inter-

relationships of these major subsystems. Prof. Atkinson was a co-principal investigator of the project concerned.

The concept of stratification has been further refined within the AG Component Engineering, and was a focus of a working group at the Fourth International workshop on Component-Based Programming [22]. The working group concluded that stratification provided a valuable and effective way of visualizing the architecture of component-based systems [17].

Prof. Atkinson, co-principle investigator of the project, is also a leader of the Kobra project at the Fraunhofer Institute for Experimental Software Engineering. This project has developed a method for component-based product line engineering using the UML [23] [24]. Explicit separation of concerns is also a key principle of the Kobra approach, and framework development and application steps are clearly divided into well-defined dimensions. In contrast with the proposed project, however, the primary focus of Kobra is the composition dimension. Kobra components are modelled by suites of UML diagrams at a level of abstraction akin to analysis/design in regular development methods, and the refinement/translation of models into code is strictly separated from (de)composition of components into subcomponents [25]. The expertise gained from Kobra will be applied to the methodological issues in the proposed project. Moreover, since it will address a different dimension of concern, it is expected that the proposed project will provide a powerful complement to the separations already achieved in the Kobra approach.

T. Kühne, the other principle investigator on the project, has contributed to establishing the limitations of inheritance in software composition [29], and formulated the “Function Object” pattern [27] which is likely to be applicable in several areas of the proposed project. Not only is the introduction of function objects a recognised way to help “white-box” frameworks mature into “black-box” frameworks, but since function objects can be of “higher-order” they can facilitate certain forms of partial parameterization. As such, function objects can serve as combinators, just as in functional programming, and may thus provide one cornerstone of a clean model of component connectors.

The function object pattern is a part of a functional pattern system [28] which T. Kühne used to describe how functional paradigm can be embedded into object-oriented development. This pattern system will not only help to identify useful architectural styles for higher-level strata [26] but will also help to assess how patterns may further be empowered by a supporting stratification approach.

## Literatur

- [20] C. W. McKay and C. Atkinson, "Supporting the Evolution of Distributed, Non-Stop, Mission and Safety Critical Systems," *Informatica*, vol. 19, no.1. 1995.

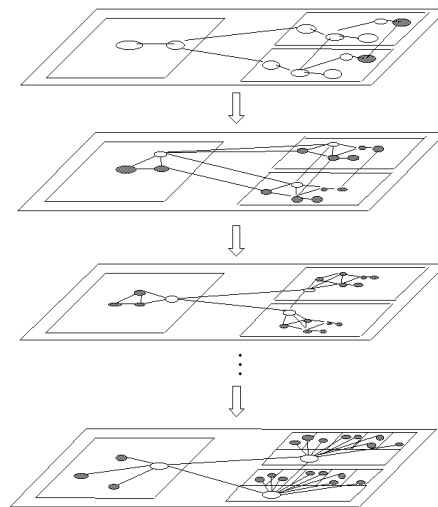
- [21] C. Atkinson and C. W. McKay, "A Generic Architecture for Distributed, Non-Stop, Mission and Safety Critical Systems," Second IFAC Workshop on Safety and Reliability in Emerging Control Technologies, Daytona Beach, FL, November 1-3.
- [22] C. Atkinson, T. Kühne, C. Bunse, Dimensions of Component-based Development, Fourth International Workshop on Component-Oriented Programming (WCOP'99; in conjunction with ECOOP'99), Lisbon, Portugal, 14 June 1999.
- [23] C. Atkinson, J. Bayer, O. Laitenberger and J. Zettel, Component-Based Software Engineering: The Kobra Approach, 2000 International Workshop on Component-Based Software Engineering, Limerick, Ireland, June 5-6, 2000.
- [24] C. Atkinson, J. Bayer and D. Muthig, Component-Based Product Line Development: The Kobra Approach, The First Software Product Line Conference, August 28-31, 2000, Denver, Colorado
- [25] C. Bunse and C. Atkinson, Improving the Quality of Object-Oriented Software: Systematic Refinement and Translation of Models to Code, Proceedings of the International Conference on Software and System Engineering and their Application, Paris, December 1999.
- [26] T. Kühne, Internal Iteration Externalized, Proceedings of 13th ECOOP, Portugal, June 1999.
- [27] T. Kühne, The Function Object Pattern, C++ Report, Vol 9 #9, pages 32--42, October 1997.
- [28] T. Kühne, A Functional Pattern System for Object-Oriented Design, Verlag Dr. Kovac, ISBN: 3-86064-770-9, July 1999.
- [29] T. Kühne, Parameterization versus Inheritance, Proceedings of TOOLS15, Australia, November 1994.

### **3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan**

#### **3.5.1 Ziele und Methoden**

The main goal of the proposed project is to investigate, elaborate and support the concept of stratification in the creation and reuse of component-based frameworks. In doing so it will draw upon the main ideas developed in previous SFB 501 subprojects, as well as other leading edge work. At its core, a stratified framework is a partially preconfigured system organized in terms of a hierarchy of related, but distinct, architectural descriptions. This is represented schematically in Figure 1.





**Abbildung 1:** Architectural strata

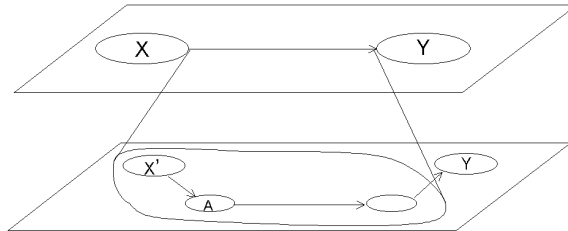
In contrast with classic “layered” architectures, which essentially group the components within a system into related bands of abstraction (i.e. layers), the strata in a stratified architecture all represent **complete** descriptions of the system. The difference between the strata lies in their component population, and the level of abstraction at which the interactions between these components is described.

The relationship between the various strata is thus one of refinement. However, the focus of the refinement is the relationships (interactions) between components rather than components themselves. The refinement of an interaction at one level generally leads not only to multiple lower-level interactions, but also to the introduction of new components. This is depicted informally in Figure 2. The same situation also occurs in formal models of component-based systems. Figure 3 shows a similar situation but in the context of the formal component model of Broy et. al. [12].

In all object-oriented refinement models, the refinement of a high-level interaction generally leads to multiple lower-level interactions plus new lower-level components. With the stratification approach, however, instead of manipulating each refinement scenario independently, as is the case in typical development methods, individual refinements are reconciled and lower level components are consolidated to yield distinct levels of architectural representation. In effect, all individual refinements are handled within the context of the overall refinement lattice.

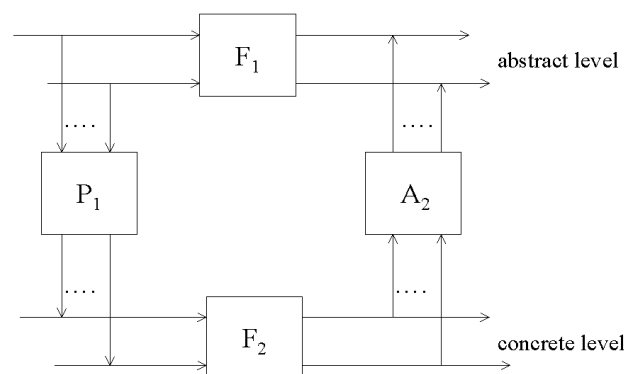
An important aspect of the proposed work is the distillation of a new form of pattern language (or system). By considering the entirety of refinement levels, one achieves a traceability between related component configurations residing at different levels which is usually lost when higher level strata are discarded. A stratified framework will thus allow the relationships between high level patterns (e.g., describing architectural styles) and other patterns at a lower level to be captured, but in a way that utilizes the typical component configuration mechanisms. Hence, investigating the

relationship of design patterns at different architectural strata will provide a way to “reverse-engineer” a pattern language from the typical, recurring application of design patterns. The rationale for this approach is the observation that proceeding down the patterns in a pattern language is akin to following the trace of a configuration of components at different architectural strata.



**Abbildung 2:** Informal model of interaction refinement

Another important research question to be addressed in the project is how to accommodate (de)composition of components within the framework. Conceptually (de)composition is an alternative dimension, but in practice it will have to be reconciled with the interaction refinement dimension and the resulting stratification levels.



**Abbildung 3:** Interaction refinement: formal model [Broy]

The systematic and coherent visualization of interaction refinement is particularly problematic in object-oriented systems since the identity of the server object is bound up in the definition of the interaction. Part of the research work will therefore involve an investigation of the difference between function-oriented and object-oriented architectures from the perspective of stratification, and where appropriate the adaptation of leading function-oriented component approaches. In particular, the principle ideas from B4, B5 and B10 will be applied in an object-oriented context.

To accomplish these goals, the proposed project will have four primary areas of work:

- a) elaboration and description of the principles of framework stratification, and the properties of a stratified framework (the product),

- b) elaboration and description of a practical method for stratified framework development and application (the process),
- c) validation of the concept of framework stratification through the development of a case study (in the area of building automation),
- d) development of “stratification” tool support for the development and application of stratified frameworks.

### 3.5.2 Arbeitsprogramm

#### **Arbeitsschwerpunkt A: Methodological Foundations of Framework Stratification**

The goal of this work area is to lay the methodological foundation for stratification and its practical use in the creation and application of frameworks. As with any engineering approach, two fundamental aspects of the method can be distinguished - the **product** and the **process**. The work area is thus divided into two work packages one focusing on the product and the other focusing on the process.

##### **Arbeitspaket A1:** Definition of the foundation principles (the product)

In this work package the concept of framework stratification will be consolidated and the key principles elaborated. The result will be a description of the product generated in a stratification project. The first activity in the work package will be the adaptation and integration of the relevant foundation technologies, especially the identified component models, architecture visualization techniques and/or patterns and interaction refinement model. Ideas from B5 and B10 are expected play an important role in this activity, as will the underlying conceptual models of leading component-based architectures and frameworks. The process will then be repeated to develop a strategy for capturing the relationships between strata. Ideas from B4 are expected to be of relevance here. Wherever possible, the approach will adopt practical, leading edge development technologies such as the UML and other OMG based-standard, such as the Object Management Architecture (OMA). Among other things, therefore, this work package will provide an object-oriented application of the key ideas from previous SFB 501 projects using important emerging object-oriented standard such as the UML and CORBA.

Personalbedarf: 12 PM

##### **Arbeitspaket A2:** Definition of the process for developing and applying Stratified Frameworks.

Once the characteristics of a stratified framework have been defined, the process for creating stratified frameworks and for using them in the creation of software systems can be defined. This is the goal of work package A2. As in the previous work package, the first activity will involve the adaptation and integration of relevant foundation technologies. The key challenge lies not so much in the definition of techniques

for elaborating the structure of an individual strata, since this is expected to rely heavily on existing patterns and architectural styles, but rather in identifying **what** strata are the most appropriate for a given domain, and how they should be inter-related. So far as possible, the process will be based on the existing SFB 501 process for software development [18]. In addition, “separate of concern” techniques such as aspect-oriented development [9], reflective architectures and extended design spaces (B10) [10] will be investigated. Techniques for framework scoping and generalization will also be adapted from well known product-line approaches such as PuLSE [16] and FAST [15]. The investigation of separation of concerns will also consider the role of component (de)composition and its relationship to architectural strata. The contribution of stratification to pattern language (system) formulation will be investigated in this work package.

Personalbedarf: 18 PM

### **Arbeitsschwerpunkt B: Case Study**

The basic goal of this work area is to validate the approach developed in work area A by means of a case study. In accordance with the general direction of SFB 501, the case study will lie in the area of building automation. Framework-oriented development of software families is broadly speaking a two step process: the first step is the generation of a generic framework which is generalized to cover at least the common needs of systems in the domain, and the second step is the instantiation of the generic framework for a specific system. Work area B is divided into two subpackages which roughly corresponding to these two activities.

#### **Arbeitspaket B1: Stratified Framework Development**

The goal of this package is to develop a prototype stratified framework. Since frameworks, by their nature, are intended to provide extensive coverage of the domain of interest (building automation in this case) it will not be possible to create a full-scale framework with a complete complement of components, or even of architectural strata. The goal of this work package is rather to build a cut-down prototype version of a framework in order to demonstrate the feasibility of the approach, and to illustrate the nature of the components and strata that will result in the stratification approach. The aim therefore, will be to define a self-contained subset of a full-scale framework which can be carried through to implementation and ultimately running code.

The key activities that will be carried out during this activity will be the analysis and scoping of the domain, the identification and consolidation of architecture strata, the selection, analysis and modelling of self-contained subset of components, and the translation of the invariant parts of these components into a suitable implementation vehicle. The first of these activities, domain analysis and scoping, will draw heavily on the domain analysis work carried previously in the initial phase of the SFB 501 [19]. The goal will be to identify a system boundary which fits closely with previous SFB 501 case-studies, and thus facilitates optimal comparison of approaches

To maximize the dissemination and potential adoption of the approach, the case study will focus on the use of leading edge, object-oriented development and execution technologies. In particular, the UML will be used at the analysis/design level, Java will be used as the programming language and JavaBeans and/or CORBA will provide the underlying component technology. As well as demonstrating/validating the principles of stratification, therefore, this project will provide a demonstration of key ideas from B5 and B10 in the context of leading edge object-oriented/ component-based development technologies. Both will require care in the selection of representative strata, and the implementation of representative components.

Personalbedarf: 18 PM

### **Arbeitspaket B2: Framework Instantiation**

The goal of this work package is to perform an illustrative instantiation of the framework to yield working code. A representative set of application features will be selected, and the corresponding components coded to provide the desired functionality. As with the development of the framework itself, the underlying goal will be to demonstrate as many of the advantages of stratification as possible, particularly the increased ease of instantiation and adaptation that is expected. This means that system behaviour and features will be selected so as to require the instantiation of components at various strata in the framework.

Personalbedarf: 12 PM

### **Arbeitsschwerpunkt C: Tool Development**

The essence of the stratification idea is the explicit storage and manipulation of entities at all meaningful levels of abstraction (much of which is often discarded today), and the explicit storage and manipulation of the relationships between these entities. In traditional software engineering methods, this information is often referred to as traceability information. The stratification approach, therefore, has much to gain from the practical support of a dedicated tool. The purpose of this work package is to develop a prototype stratification tool. Three main work packages are envisaged for this task: Requirements Specification, Meta-Model Development and Software Development

#### **Arbeitspaket C1: Requirements Specification**

The goal of this work package is to define the requirements for a stratification tool, first for an ideal tool, and second for the specific prototype to be constructed during the project. The requirements specification will include definitions of the functional requirements, the non-functional requirements and the logical interface. A mixture of textual documents and UML diagrams will be used.

Personalbedarf: 12 PM

**Arbeitspaket C2: Meta Model Development**

Since the tool is essentially an information processing tool, a metamodel will be defined to capture the essence of the concepts to be stored and manipulated by the tool. This meta-model will not only serve as the basis for the implementation of the tool, but will also help provide a concise, abstract description of the key concepts of stratification. Established UML-oriented techniques for metamodel development and documentation will be utilized.

Personalbedarf: 10 PM

**Arbeitspaket C3: Tool Construction**

In the final work package a working prototype will be generated. A standard three-phase approach will be adopted, involving the development of an architecture, a detailed design and code. UML will be used for the description of the architecture and the detailed design, and Java will be used as the implementation language.

Personalbedarf: 8PM

**3.5.3 Zeitplan**

| Personal              | 2001<br>1. H. J. | 2001<br>2. H.J. | 2002<br>1. H.J. | 2002<br>2. H.J. | 2003<br>1. H.J. | 2003<br>2. H.J. |
|-----------------------|------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Grundausstattung      |                  |                 |                 |                 |                 |                 |
| Kühne (20 Std.)       | A1               |                 | A2              |                 | B1              | B2              |
| Ergänzungsausstattung |                  |                 |                 |                 |                 |                 |
| NN (38,5 Std.)        | A1               | A2              |                 | B1              | B1/B2           | C2              |
|                       |                  |                 |                 |                 |                 |                 |
| HW1 (19 Std.)         | B1               | B2              | C1              |                 | C2/C3           | C3              |
|                       |                  |                 |                 |                 |                 |                 |
| HW2 (19 Std.)         | B1               | B2              | C1              |                 | C2/C3           | C3              |
|                       |                  |                 |                 |                 |                 |                 |

**3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs**

The central goal of the SFB 501 is the investigation of generic techniques for the cost-effective engineering of large software systems. This project focuses directly on this goal in the context of mainstream, leading-edge, object-oriented technologies; namely components, patterns, architectures and the UML. If successful, the developed approach will offer a way of combining the advantages of these individual technologies to better support reuse. Moreover, the larger the system, the greater the expected benefits.

The proposed project will build upon ideas developed in the first round subprojects in SFB 501, namely subprojects B4, B5 and B10. As well as integrating results from

these projects, therefore, the project will also apply some of their key ideas in the context of leading-edge object-oriented methods and component-based platforms. The proposed project will also build upon the standard SFB 501 software development process, and will reuse prior work on the analysis of the SFB 501 case study domain.

Given the increased use of distributed systems, communication support is likely to play an important role in most large software systems. From the perspective of stratified architectures, this means that at least one of the levels in a stratified framework is likely to be oriented towards the description of the communication subsystem. Communication architectures developed within the subproject B4 will serve as the basis for the communication level architectural descriptions which will be needed in the case study. Also, the “protocol” patterns developed in B4 are likely to be valuable in developing the interaction refinements at (or close to) the communication stratum.

The underlying paradigm under investigation in the project is the component paradigm. This means that each architectural stratum essentially represents a component-based architecture focusing on a certain aspect of a system’s realization. The organization and description of each architectural level will therefore draw on the architectural concepts and generic component concepts developed in subproject B5.

The key to the successful stratification of frameworks is the careful identification, architecting and positioning of the different strata in the framework. This essentially involves the crystallization of the optimal dimensions of concern for the system(s) in hand. Several development paradigms address this issue from different angles, but one of the potentially most effective is the design spaces concept elaborated within the subproject B10. In particular, the notion of extended design spaces will be investigated as a strategy for identifying and structuring architectural strata. Whereas B10 focuses on the use of design spaces for the construction of a single architecture, however, B12 will investigate the approach’s potential for supporting the definition of multiple architecture visualizations, and for explicitly defining the nature of the relationships between them.

### 3.7 Ergänzungsausstattung für das Teilprojekt

|    | 2001                               |      |              | 2002                               |      |              | 2003                               |      |              |
|----|------------------------------------|------|--------------|------------------------------------|------|--------------|------------------------------------|------|--------------|
|    | Verg.-Gr.                          | Anz. | Betrag in DM | Verg.-Gr.                          | Anz. | Betrag in DM | Verg.-Gr.                          | Anz. | Betrag in DM |
| PK | BAT IIa                            | 1    | 104.400      | BAT IIa                            | 1    | 104.400      | BAT IIa                            | 1    | 104.400      |
|    | StdHkr.                            | 2    | 40.800       | StdHkr.                            | 2    | 40.800       | StdHkr.                            | 2    | 40.800       |
|    |                                    |      |              |                                    |      |              |                                    |      |              |
|    | zus.:                              | 3    | 145.200      | zus.:                              | 3    | 145.200      | zus.:                              |      | 145.200      |
| SV | Kostenkategorie oder Kennziffer    |      | Betrag in DM | Kostenkategorie oder Kennziffer    |      | Betrag in DM | Kostenkategorie oder Kennziffer    |      | Betrag in DM |
|    | (522)                              |      | 5.000        | (522)                              |      | 3.000        | (522)                              |      | 3.000        |
|    |                                    |      |              |                                    |      |              |                                    |      |              |
|    | zusammen                           |      | 5.000        | zusammen                           |      | 3.000        | zusammen                           |      | 3.000        |
| I  | Mittel für Investitionen insgesamt |      |              | Mittel für Investitionen insgesamt |      |              | Mittel für Investitionen insgesamt |      |              |
|    | -                                  |      |              | -                                  |      |              | -                                  |      |              |

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

SV: Sächliche Veraulungsausgaben (Begründung vgl. 3.7.2)

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



## 3.7.1 Begründung des Personalbedarfs

|                                                                            | Name, akad. Grad,<br>Dienststellung                              | engeres<br>Fach des<br>Mitarbeiters       | Bezeichnung des Institutes<br>oder der Hochschule     | Anteil der<br>aufgew.<br>Gesamt-<br>arbeitszeit<br>(bera-<br>tend: B) | im<br>SFB<br>tätig<br>seit | derzeitige<br>Einstufung<br>und<br>beantragte<br>Ver-<br>gütungs-<br>gruppe |
|----------------------------------------------------------------------------|------------------------------------------------------------------|-------------------------------------------|-------------------------------------------------------|-----------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------------------|
| Grundausrüstung<br>3.7.1.1<br>wiss. Mitarbeiter<br>(einschl. StdHkr.)      | (1) Atkinson, C.; Prof. Dr.<br>(2) Kühne, K., Dr.                | Prakt. Inf.<br>Prakt. Inf.                | FB Inf., Uni KL<br>FB Inf., Uni KL                    | 5<br>20                                                               |                            | C3<br>IIa                                                                   |
| Ergänzungsausrüstung<br>3.7.1.3<br>wiss. Mitarbeiter<br>(einschl. StdHkr.) | (1) N.N., Dipl-Inform.<br>(2) N.N., StdHkr.<br>(3) N.N., StdHkr. | Prakt. Inf.<br>Prakt. Inf.<br>Prakt. Inf. | FB Inf., Uni KL<br>FB Inf., Uni KL<br>FB Inf., Uni KL | 38,5<br>19<br>19                                                      |                            | IIa                                                                         |
| 3.7.1.4<br>nichtwissenschaftl.<br>Mitarbeiter                              |                                                                  |                                           |                                                       |                                                                       |                            |                                                                             |

### Aufgabenbeschreibung für Mitarbeiter der Grundausrüstung

T. Kühne will work on the definition of the foundation principles of the stratified framework concept and will work on the development of a process to create and apply stratified frameworks. He will furthermore, contribute to and consolidate the results of developing a prototype stratified framework for a domain fitting closely with previous SFB 501 case-studies.

### Aufgabenbeschreibung für Mitarbeiter der Ergänzungsausrüstung

This Mitarbeiter will contribute to the definition of the process for developing and applying Stratified Frameworks. and then focus on the validation of the results achieved in work area A by means of a case study. He will also contribute to the conceptual foundations for a tool design.

Both studentische Hilfskräfte will perform preparatory work for workpackage B and then focus their attention to the conception and implementation of a prototype stratification tool.

### 3.7.2 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben

|                                                                                                                                                                | 2001                                                                                                                                                                                   | 2002  | 2003  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|
| Für sächliche Verwaltungsausgaben stehen als <b>Grundausrüstung</b> voraussichtlich zur Verfügung                                                              | Mittel für sächliche Verwaltungsausgaben in der Grundausrüstung werden zentral vom Fachbereich Informatik verwaltet und können deshalb nicht in Arbeitsgruppen aufgeschlüsselt werden. |       |       |
| Für sächliche Verwaltungsausgaben werden als <b>Ergänzungsausrüstung</b> beantragt<br>(entspricht den Summen "sächliche Verwaltungsausgaben" in Übersicht 3.7) | 5.000                                                                                                                                                                                  | 3.000 | 3.000 |

### Begründung zur Ergänzungsausrüstung der Sächlichen Verwaltungsausgaben

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

### 3.7.3 Investitionen (Geräte über 20.000 brutto)

| Bezeichnung des Gerätes<br>(ggf. mit Typenbezeichnung) | Einzelpreis<br>2001 | Einzelpreis<br>2002 | Einzelpreis<br>2003 |
|--------------------------------------------------------|---------------------|---------------------|---------------------|
|                                                        | -                   | -                   | -                   |
| Summe                                                  | -                   | -                   | -                   |