Entwicklung großer Systeme mit generischen Methoden

Arbeits- und Ergebnisbericht

1998 - 1999 - 2000

Sonderforschungsbereich 501

Universität Kaiserslautern

April 2000

Sonderforschungsbereich 501

Entwicklung großer Systeme
mit generischen Methoden

Zusammensetzung des Vorstands 1998 - 2000:

Sprecher:	Prof. Dr. Jürgen Nehmer
stellv. Sprecher:	Prof. Dr. Dieter Rombach
Vorstand:	Prof. Dr. Jürgen Avenhaus
	Prof. Dr. Reinhard Gotzhein
	Prof. Dr. T. Härder
	Prof. Dr. L. Litz
	Prof. Dr. Klaus Madlener
	Prof. Dr. Jürgen Nehmer
	Prof. Dr. Michael Richter
	Dr. Norbert Ritter
	Prof. Dr. Dieter Rombach
	Dr. Bernd Schürmann
	Prof. Dr. Gerhard Zimmermann

Inhaltsverzeichnis

1 Allgemeiner Teil

1.1 Wissenschaftliche Entwicklung des SFB

Die Teilprojekte des SFB 501 haben sich in der zweiten Förderperiode wissenschaftlich gut entwickelt, was in der stark angestiegenen Publikationsleistung zum Ausdruck kommt. Im Februar 1999 wurde vom SFB ein Workshop über das Thema "Software Reuse" unter Beteiligung namhafter deutscher und ausländischer Wissenschaftler an der Universität Kaiserslautern ausgerichtet. Fachliche Verstärkung erhielt das Arbeitsgebiet des SFB durch die von der Universität und dem Fraunhofer-Institut IESE gemeinsam finanzierte C3-Professur "Component Engineering", die erfolgreich mit Herrn Prof. Atkinson besetzt werden konnte. Herr Atkinson wird auch den SFB in der kommenden Förderperiode um ein Teilprojekt bereichern.

Auf dem Anwendungsgebiet des SFB hat sich der Fachbereich durch eine C3-Professur (Prof. Henrich, Arbeitsgebiet Eingebettete Systeme und Robotik) verstärkt. Herr Schürmann, der seit 1.7.1998 eine C2-Professur für Entwurfsmethodik eingebetteter Systeme im Fachbereich bekleidet, wird in der neuen Förderperiode ebenfalls mit einem eigenständigen Teilprojekt vertreten sein und sich um die nichtfunktionale Eigenschaft der Zuverlässigkeit besonders kümmern.

Die Kontakte zu in- und ausländischen Wissenschaftlern haben sich - ausgelöst durch die Teilnahme von SFB-Wissenschaftlern an zahlreichen Tagungen - erfreulich gut entwickelt (siehe auch 2.1).

Im Berichtszeitraum konnte ferner der Kontakt zur Industrie verstärkt werden. Einige Teilprojekte sind mit Firmen bereits in konkreten Verhandlungen über die Erprobung von SFB-Methoden in industriellen Softwareprojekten. Es besteht die Absicht zur Einrichtung eines Transferbereichs zum Jahresbeginn 2001. Über den konkreten Stand dieser Planungen wird anlässlich der Begehung berichtet.

1.2 Entwicklung der Kooperation

Die SFB-interne Kooperation wurde auf der Basis von Querschnittprojekten fortgeführt, da sich diese Organisationsform in den abgelaufenen Förderperioden sehr bewährt hat. Querschnittprojekte werden für teilprojektübergreifende Untersuchungen und Experimente ad-hoc eingerichtet, wenn sich die Notwendigkeit dafür ergibt. Dazu entsenden Teilprojekte je nach Bedarf Mitarbeiter mit ca. 8 Std./Woche in Teams, die die Querschnittprojekte bearbeiten und sich danach wieder auflösen. In der abgelaufenen Förderperiode wurden folgende Querschnittprojekte eingerichtet:

Q1: Prototyping eines Gebäudeautomationssystems
In diesem Querschnittprojekt wurden zwei aufeinander aufbauende kontrollierte Entwicklungsprojekte durchgeführt, deren Ziel die Erstellung ausführbarer Anforderungen in SDL war.

Q2: Entwicklung eines Gebäudeautomationssystems
In diesem Querschnittprojekt wurde ein Entwicklungsprojekt mit Studenten durchgeführt, das die Phasen Entwurf und Kodierung einschloss.

Q3: Erarbeitung eines Domänenmodells
Dieses Querschnittprojekt hatte die Aufgabe, unter Rückgriff auf etablierte Methoden des Domain-Engineering das vorhandene Wissen über das Anwendungsfeld Gebäudeautomation zu strukturieren und als Basis einer systematischen Wiederverwendung in ein Domänenmodell zu gießen.

Q4: Anforderungsanalyse eines Gebäudeautomationssystems
Im Rahmen dieses Querschnittprojektes wurden aufbauend auf den Ergebnissen der Kooperation der Teilprojekte B4 und C1 mehrere natürlich-sprachliche Problembeschreibungen aus dem Anwendungsfeld Gebäudeautomation in eine formale, objektorientierte Problemspezifikation überführt. Die Ziele dabei waren zum einen, allgemein die Eignung des entwickelten Ansatzes für dieses Anwendungsfeld zu überprüfen, und zum anderen, speziell die Möglichkeiten der Wiederverwendung von Teilen einer Problemspezifikation zu untersuchen.

Die Ergebnisse der Querschnittprojekte erscheinen in diesem Bericht als eigenständige Berichtsteile.

Mit dem Fraunhofer-Institut für Experimentelles Software Engineering (Leiter Prof. Rombach) bestehen zahlreiche Verflechtungen, z. B. bei der Mitarbeit von SFB-Kollegen an Projekten des Instituts. Wir betrachten die Kooperation mit dem IESE als eine Chance, Ergebnisse der Grundlagenforschung schnell in die industrielle Praxis zu transferieren.

1.3 Stellung innerhalb der Hochschule

Der SFB 501 ist z. Zt. der einzige Sonderforschungsbereich an der Universität Kaiserslautern und hat eine dementsprechend herausragende Stellung. Die Thematik des Sonderforschungsbereichs wird die im Zuge von Stellenwiederbesetzungen angestrebte Profilierung des Fachs Informatik an der Universität Kaiserslautern entscheidend beeinflussen. Der Fachbereich Informatik hat sich entschlossen, die ingenieurmäßige Konstruktion komplexer Softwaresysteme in das Zentrum seiner zukünftigen Anstrengungen in Forschung und Lehre zu stellen. Entsprechende Planungen sind bereits mit der Hochschulleitung abgesprochen.

1.4 Förderung der Lehre und des wissenschaftlichen Nachwuchses

Angeregt durch Arbeiten des SFB wurde im Wintersemester 1998/99 ein neuer Studiengang in Angewandter Informatik an der Universität Kaiserslautern eingerichtet. Bereits die Einführungsveranstaltungen (Softwareentwicklung I - III) betonen den ganzheitlichen Prozess der Softwareentwicklung und stellen in dieser Hinsicht ein Novum in der Grundausbildung dar. Ein weiterer Aspekt ist die frühe Integration eines Anwendungsfaches bereits im dritten Semester, wobei die Vertiefungen "betriebliche Informationssysteme" und "eingebettete Systeme" gewählt werden können.

2 Dokumentation der sonstigen Aktivitäten

2.1 Kontakte mit anderen Wissenschaftlern

Prof. J. Avenhaus: Kooperation mit Prof. Küchlin, Universität Tübingen, auf dem Gebiet des verteilten Beweisens

Prof. J. Avenhaus: Kooperation mit Prof. Walther, TU Darmstadt, auf dem Gebiet des Induktionsbeweisens

Prof. J. Avenhaus: Kooperation mit Prof. Plaisted, University of North Carolina, USA, auf dem Gebiet des Theorembeweisens

Prof. R. Gotzhein: Kooperation im Bereich Requirements Engineering mit Prof. Börge, Universität Pisa, Italien

Prof. R. Gotzhein: Kooperation im Bereich SDL-Semantik mit Dr. Glässer, Universität Paderborn, und Dr. Prinz, Humboldt-Universität Berlin

Prof. T. Härder: Entwicklung objekt-relationaler DBMS mit dem DBTI der IBM San Jose, USA (Dr. Deßloch, Dr. Mattos)

Prof. T. Härder: Kooperativer Entwurf und Versionierung mit Prof. Iochpe, Instituto de Informatica, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasilien

Prof. T. Härder: Middleware-Technologie mit dem Forschungsinstitut Ulm der Daimler Chrysler AG (Dr. Sauter, Dr. Sellentin)

Prof. T. Härder: Infrastruktur heterogener Informationssysteme und DB-Middleware mit Prof. Mitschang, Universität Stuttgart

Prof. L. Litz: Kooperation zum Thema mathematische Modellbildung thermischen Gebäudeverhaltens mit dem Institut für Energietechnik, FB Verfahrenstechnik, Umwelttechnik, Werkstoffwissenschaften der TU Berlin (Dipl. Ing. C. Nytsch)

Prof. L. Litz: Kooperation zum Thema objektorientierte mathematische Modellbildung mit Dr. M. Otter, Deutsches Zentrum für Luft- und Raumfahrt, Wessling

Prof. L. Litz: Kooperation mit Prof. H. Heinrich, Lehrstuhl für Bauphysik der Universität Kaiserslautern im Bereich thermischer Simulationen und Validierung mathematischer Modelle

Prof. L. Litz: Kooperation zum Thema Regelungskonzepte von solarbeheizten Niedrigenergiehäusern mit dem Fraunhofer-Institut für solare Energiesysteme, Freiburg

Prof. L. Litz: Kooperation mit der Ecole Normale Superieure de Cachan, Laboratoire Universitaire de Recherche en Production Automatisee, Cachan Cedex, Frankreich (Dr. Lampriere-Couffin) zum Thema "formale Methoden zum Steuerungsentwurf"

Prof. K. Madlener: Kooperation mit Prof. Cardoso und Prof. Casallas, Universidad de los Andes, Bogota, Kolumbien, im Bereich Softwareentwicklung

Prof. K. Madlener: Kooperation mit Prof. Otto, Gesamthochschule Kassel, auf dem Gebiet Computer-Algebra und Komplexität

Prof. K. Madlener: Kooperation mit Prof. Narendran, SUNY, Albany, N.Y., USA, auf dem Gebiet der Komplexität

Prof. J. Nehmer: Kooperation mit Prof. Karshmer, New Mexico State University, Las Cruces, NM, USA (verbrachte 1998 sein Sabbatical im SFB)

Prof. J. Nehmer: Kooperation mit Prof. Heiser, New South Wales University, Sydney, Australien auf dem Gebiet der Ein-Adressraum-Betriebssysteme (s. Kontaktreise Januar/Februar 1998)

Prof. J. Nehmer: Kooperation mit Prof. Magee, Imperial College, London, UK, auf dem Gebiet der nebenläufigen Systeme

Prof. J. Nehmer: Kooperation mit Prof. Jarzabek, National University of Singapore, auf dem Gebiet Softwarearchitekturen

Prof. J. Nehmer: Kooperation mit Prof. Mattern, Eidgenössische Technische Hochschule, Zürich, Schweiz, auf dem Gebiet Ubiquitous Computing

Prof. J. Nehmer: Kooperation mit Prof. Rammig, Universität Paderborn, auf dem Gebiet der generischen Betriebssysteme

Prof. J. Nehmer: Kooperation mit Prof. Schröder-Preikschat, Universität Magdeburg, auf dem Gebiet der generischen Betriebssysteme

Prof. M. Richter: Kooperation mit Prof. Maurer, University of Calgary, Canada zur Weiterentwicklung des MILOS-Systems (Förderung durch das BMBF, Projekt-Nr. KAN99/007 INF)

Prof. M. Richter: Kooperation mit Dr. Petrie, Stanford University, USA. Im Rahmen dieser Kooperation wurden die in der Software-Engineering-Domäne entwickelten Konzepte zur verzahnten Planung und Ausführung von Projekten auf Projekte im Beresich Maschinenbau übertragen

Prof. D. Rombach: Kooperation zum Thema Software-Inspektionen, Messen von Software, Methodenanalyse mit der Software Engineering Group (Prof. Basili) der University of Maryland, USA

Prof. D. Rombach: Kooperation zum Thema Prozessmodellierung und Prozessdokumentation mit Prof. Kellner, Carnegie Mellon University, Pittsburgh, USA

Prof. D. Rombach: Kooperation zum Thema "NRL-Spezifikationsansatz" mit Prof. Parnas, McMasters University, Hamilton, Ontario, Kanada

Prof. D. Rombach: Kooperation zum Thema "Requirements Analysis" mit Prof. Leite, Pontificia Universidade Catolica do Rio de Janeiro, Brasilien

Prof. D. Rombach: Kooperation zum Thema "Allgemeine Software Engineering Ansätze" mit dem Nara Institute of Science and Technology, Japan (Prof. K. Torii, Prof. H. Iida)

Prof. D. Rombach: Kooperation zum Thema "Statistisches Testen auf Anforderungsebene mit Prof. Poore, University of Tennessee, USA

Prof. G. Zimmermann: Kooperation im Bereich Gebäudeautomation mit Prof. Hartkopf, Carnegie Mellon University, USA

Prof. G. Zimmermann: Kooperation im Bereich Gebäudeautomation und Simulation mit Prof. Mahdavi, Carnegie Mellon University, USA

2.2 Mitwirkung an Kongressen und Fachtagungen

Prof. J. Avenhaus: PC-Mitglied der "9th International Conference on Tewriting Techniques and Applications", Tsukuba, Japan, 1998

Prof. R. Gotzhein: Exponat "Ingenieurmäßige Entwicklung maßgeschneiderter Kommunikationssoftware" auf der CeBIT'98, Hannover, März 1998

Prof. R. Gotzhein: Exponat "Maßschneiderung echtzeitfähiger Kommunikationssoftware" auf der CeBIT'98, Hannover, März 1998

Prof. R. Gotzhein: Steering Committee Member + Organisation und Leitung der Sitzungen "Patterns and SDL" des internationalen Workshops "SDL and MSC" (SAM'98), Berlin, Juni 1998

Prof. R. Gotzhein: Program-Co-Chair des internationalen Workshops "The Formal Description Technique Estelle" (Estelle'98), Evry, Frankreich, November 1998

Prof. R. Gotzhein: PC-Mitglied der internationalen IFIP-Konferenz "Formal Description Techniques/Protocol Specification, Testing, and Verification" (FORTE/PSTV'98), Paris, Frankreich, November 1998

Prof. R. Gotzhein: PC-Mitglied des internationalen Workshops "Advanced Intelligent Networks" (ACoS'98) in Lissabon, Portugal, April 1998

Prof. R. Gotzhein: PC-Mitglied der internationalen IFIP-Konferenz "Formal Description Techniques / Protocol Specification, Testing, and Verification" (FORTE/ PSTV'98), Beijing, China, Oktober 1999

Prof. R. Gotzhein: PC-Mitglied der internationalen IFIP-Konferenz "Formal Methods for Open Object-Based Distributed Systems" (FMOODS'99), Florenz, Italien, Februar 1999

Prof. R. Gotzhein: PC-Mitglied der internationalen Konferenz "Colloque Francophone sur l'Ingenierie des Protocoles" (CFIP'99), Nancy, Frankreich, April 1999

Prof. R. Gotzhein: PC-Mitglied der internationalen IFIP-Konferenz "Formal Description Techniques / Protocol Specification, Testing, and Verification" (FORTE/ PSTV'2000), Pisa, Italien, Oktober 2000

Prof. R. Gotzhein: PC-Mitglied der internationalen Konferenz "Colloque Francophone sur l'Ingenierie des Protocoles" (CFIP'2000), Toulouse, Frankreich, Oktober 2000

Prof. T. Härder: PC-Mitglied der 23th Conference on Very Large Data Bases (VLDB), Athen, Griechenland, Aug. 1997

Prof. T. Härder: PC-Mitglied der 12th Brazilian Symposium on Data Bases (SBBD'97), Fortaleza, Brasilien, Okt. 1997

Prof. T. Härder: PC-Mitglied der GI-Fachtagung "CAD'98 - Tele-CAD: Produktentwicklung in Netzwerken", Darmstadt, März 1998

Prof. T. Härder: PC-Mitglied der ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'98), Seattle, WA, USA, Mai 1998

Prof. T. Härder: PC-Mitglied des Int. Workshop on New Database Technologies (together with ER'98), Singapore, Nov. 1998

Prof. T. Härder: PC-Mitglied der GI-Fachtagung "Datenbanken in Büro, Technik und Wissenschaft" in Freiburg, März 1999

Prof. T. Härder: PC-Mitglied der 25th Int. Conference on Very Large Data Bases (VLDB), Edinburgh, UK, Aug. 1999

Prof. T. Härder: PC-Mitglied der GI-Fachtagung "CAD'2000", Berlin, März 2000

Prof. T. Härder: Vice Chair der International Conference on Data Engineering, San Diego, CA, USA, März 2000

Prof. T. Härder: PC-Mitglied der 7th Int. Conference on Extending Database Technology (EDBT2000), Konstanz, März 2000

Prof. T. Härder: PC-Mitglied der Int. Conference on Web Information System Engineering (WISE'2000), Hong Kong, Juni 2000

Prof. T. Härder: PC-Mitglied des 10th Int. Database Symposium on Mobile, XML and Postrelational Databases (IDS2000), Hong Kong, Juni 2000

Prof. M. Richter: Forschungsforum Wissensmanagement und schnelle Produktentwicklung, Stuttgart, November 1999

Dr. N. Ritter: PC-Mitglied der Int. Conf. Euro-Par 2000 (Parallel and Distributed Databases and Applications), München, Aug. 2000

Prof. D. Rombach: Mitglied Steering Commitee, International Conference on Software Engineering (ICSE), Kyoto; Japan, April 1998 und Los Angeles, CA, USA, Mai 1999

Prof. D. Rombach: General Chair, International Conference on Software Engineering and Knowledge Engineering (SEKE '99), Kaiserslautern, Juni 1999

Prof. D. Rombach: General Chair, Product Focussed Improvement for Embedded Software Processes (Profes '99), Oulu, Finnland, Juni 1999

Prof. D. Rombach: PC-Mitglied, Sixth International Symposium on Software Metrics '99, Boca Raton, Florida, USA, November 1999

Dr. B. Schürmann: Topic-Chair der Int. Konferenz Design Automation and Test in Europe (DATE), Paris, Frankreich, 1998

Dr. B. Schürmann: PC-Mitglied der Int. Konferenz Design Automation and Test in Europe (DATE), München, 1999

Prof. G. Zimmermann: PC-Mitglied der Int. Konferenz New Information Technologies in Science, Education, Telecommunications and Business (IT + SE), Yalta-Gurzuff, Ukraine, 1998

Prof. G. Zimmermann: PC-Mitglied der Int. Konferenz Design Automation and Test in Europe (DATE), Paris, 2000

2.3 Kongress-, Vortrags- oder Kontaktreisen nach Übersee oder Osteuropa von Angehörigen des Sonderforschungsbereichs

2.3.1 Kongress- und Vortragsreisen

L. Baum: International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Las Vegas, Nevada, USA, Juni 1999. Reisekostenzuschuss: 2.948,30 DM

L. Baum: 4th International Conference on Requirements Engineering (ICRE-2000), Schaumburg, Illionois, USA, Juni 2000

R. Feldmann: 10th International Conference on Software Engineering and Knowledge Engineering (SEKE ’98), San Francisco, USA, Juni 1998. Reisekostenzuschuss: 3.154,34 DM

B. Geppert: International Conference on Software Reuse, Victoria, British Columbia, Canada, Mai/Juni 1998. Reisekostenzuschuss: 3.609,82 DM

S. Goldmann und H. Holz: IEEE 7th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’98) Stanford University, Palo Alto, CA, USA, Juni 1998. Reisekostenzuschuss: Goldmann 1.564,00 DM, Holz 3436,00 DM

S. Goldmann und B. Kötting: IEEE 8th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE ’99) Stanford University, Palo Alto, CA, USA, Juni 1999. Reisekostenzuschuss: 2.265,06 DM

Prof. R. Gotzhein: IEEE International Conference on Formal Engineering Methods, Hiroshima, Japan, Nov. 1997. Reisekostenzuschuss: 4.645,10 DM

Prof. R. Gotzhein: Vortragsreise Osaka University, Sendai University, Yamagata University, Japan, Nov. 1997. Reisekostenzuschuss: s.o.

H. Holz: 21st International Conference on Software Engineering (ICSE-99), Los Angeles, USA, Mai 1999. Reisekostenzuschuss: 3.757,50 DM

Prof. K. Madlener: Rewriting Techniques and Applications (RTA ’98), Tsukuba, Japan, März 1998. Reisekostenzuschuss: 2.701,10 DM

G. Molter und J.P. Riegel: First Working IFIP Conference on Software Architecture (WICSA1), San Antonio, Texas, USA, Februar 1999. Reisekostenzuschuss: 2.505,16 und 2.215,07 DM

Prof. J. Nehmer: Vortragsreise an die New South Wales University, Australien, Januar/ Februar 1998. Aufenthalt von 6 Wochen mit folgenden Vorträgen: "Development of Large Systems with Generic Methods", "Customization of Operating Systems - A Generic Approach", "Software Development for Application Domains - A Product Centered Approach"

Dr. N. Ritter: Information Resources Management Association International Conference IRMA’99, Hershey, PA, USA, Mai 1999. Reisekostenzuschuss: 2.621,95 DM

F. Rößler: 5th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS’99), Las Vegas, Nevada, USA, Oktober 1999. Reisekostenzuschuss: 2.057,16 DM

H.P. Steiert: Third International Software Architecture Workshop (ISAW-3), Orlando, Florida, USA, November 1998. Reisekostenzuschuss: 2.980,78 DM

H.P. Steiert: OOPSLA'99-Workshop "Rigourous Modeling and Analysis with the UML: Challenges and Limitations", Denver, Colorado, USA, November 1999. Reisekostenzuschuss: 3.242,73 DM

Dr. B. Schürmann und Prof. G. Zimmermann: Int. Konferenz New Information Technologies in Science, Education, Telecommunications and Business (IT + SE), Yalta-Gurzuff, Ukraine, Mai 1998. Reisekostenzuschuss: 2.323,16 DM und 2.348,41 DM

Prof. G. Zimmermann: SLIP99-Workshop on System-level Interconnect Prediction, Monterey, CA, USA, April1999

Prof. G. Zimmermann: Int. Symposium on Physical Design ISPD-99, Monterey, CA, USA, April 1999

2.3.2 Kontaktreisen

B. Dellen: Universität Calgary und Banff, Kanada, Dez. 97 - März 98. Entwicklung des Designs für das in A2 zu entwickelnde System, Betreuung von Diplomarbeiten für die Teilprojekte A2 und B2, Erarbeitung eines gemeinsamen Artikels für die Zeitschrift "International Journal of Applied Software Technology" über B2-Inhalte, Teilnahme am KAW-98 (Eleventh Workshop on Knowledge Acquisition, Modeling and Management), Banff. Reisekostenzuschuss: 3.277,94 DM

S. Goldmann: Universität Stanford (Dr. Petrie, Prof. Cutkosky, Center for Design Research). Kontaktreise im Anschluss an WETICE'98-Konferenz, Juni/Juli 1998. Arbeit an einem gemeinsamen Paper. Reisekostenzuschuss: s. WETICE'98-Konferenz

H. Holz: Universität Calgary, Kanada, April/Mai 1999. Kontaktreise. Zweck: Fertigstellung eines Prototyps für eine System-Demonstration auf der 21st ICSE-99-Conference, System-Integrationstests, Erstellen von Testreports, Redesign und Reimplementierung von Benutzungsoberflächen, gemeinsame Veröffentlichung "Knowledge Management for Learning Software Organization" für den KAW-99 (Workshop on Knowledge Acquisition, Modeling and Management). Reisekostenzuschuss: s. ICSE-Konferenz

Prof. K. Madlener: Kyoto Sangyo University, Japan, März/April 1998. Kontaktreise. Zweck: Gastaufenthalt bei Prof. Ito, Kyoto und Prof. Kobayashi an der Toho University Funabashi. Reisekostenzuschuss: s. RTA'98-Konferenz

Prof. K. Madlener: Universidad de los Andes, Bogota, Kolumbien, 1998, 1999 und 2000. Gastaufenthalte, Vorträge, Betreeung von Diplomanden, Abnahme von Prüfungen (finanziert durch DAAD)

Prof. G. Zimmermann: Carnegie Mellon University, Pittsburgh/USA, Dept. of Architecture 1997 und 1998, Kontaktreisen. Zweck: Analyse des Intelligent Workplace als komplexes Experimentierfeld für die Modellierung reaktiver Systeme in der Gebäudeautomation. Untersuchung der Nutzbarkeit eines integrierten Gebäudesimulators SEMPER, für unsere Prototyping-Zwecke. Pflege der Kontakte mit Professoren des Departments, wissenschaftlicher Austausch, Initiierung einer gemeinsamen Promotion an der CMU. Ergebnis: Weiterführung der engen Kooperation, neue domänenspezifische Analysemethode für integrierte Gebäudesteuerungen, verbesserte Modellarchitektur. Reisekostenzuschuss 1997: 1.343,20 DM / 1998: 4.793,20 DM

A. Metzger: Carnegie Mellon University, Pittsburgh/USA, Dept. of Architecture 1999, Kontaktreise. Zweck: Planung und Implementierung eines Interface zwischen der SEMPER-Umgebung und unserer auf SDL und SDT basierten Prototyping-Umgebung auf CORBA Basis. Betreuung einer Projektarbeit eines Studenten aus dem D1 Projekt über die Modellierung einer Lichtsteuerung im Intelligent Workplace. Ergebnis: Funktionierendes Interface, SEMPER in D1 installiert

2.4 Sonstiges

SFB-Jahrestreffen Juni 1998, Europäische Akademie Otzenhausen

SFB-Seminar "Middleware: Betriebssysteme der Zukunft?", Juli 1998

Internationaler SFB-Workshop "Software Reuse", Universität Kaiserslautern, Februar 1999

SFB-Kolloquium Oktober 1999, Europäische Akademie Otzenhausen

Prof. T. Härder: Mitglied des Editorial Advisory Board bei Information Systems

Prof. T. Härder: Hauptherausgeber der Zeitschrift "Informatik - Forschung und Entwicklung"

Prof. T. Härder: Herausgeber der Reihe "Datenbanksysteme" im Vieweg-Verlag

Prof. T. Härder: DFG-Fachgutachter "Praktische Informatik"

Prof. T. Härder: Sprecher des Fachausschusses 2.5 der GI "Rechnergestützte Informationssysteme"

Prof. J. Nehmer: Mitherausgeber des Informatik-Spektrum

Prof. J. Nehmer: Mitglied von Senat und Hauptausschuß der DFG

Prof. J. Nehmer: Mitglied des Wissenschaftsrates (bis Juni 1998)

Prof. D. Rombach: Mitherausgeber folgender Zeitschriften: IEEE Software Magazine; The Journal of Systems and Software; Informatik: Forschung und Entwicklung;

International Journal of Software Process: Improvement and Practice; International Journal of Empirical Software Engineering

2.5 Gastwissenschaftler im SFB

1998

Prof. Alfs Berztiss
University of Pittsburgh, USA
5.1. - 19.1.98

Prof. Oleg Vityaz
Universität Kiew, Ukraine
13.1. - 27.1.98

Dr. Stefan Fischer
Universität Mannheim
11.2.98

Prof. F. Wysotzki
Technische Universität Berlin
22.3. - 25.3.98

Dr. Charles Petrie
University of Stanford, California / USA
26.5. - 29.5.98
8.7. - 18.7.98

Prof. Hajimu Iida
Nara Institute of Science and Technology, Nara / Japan
1.6. - 30.11.98

Prof. Arthur Karshmer
New Mexico State University, Las Cruces, NM / USA
1.6. - 31.7.98

Dr. Burkhard Stiller
ETH Zürich, Schweiz
6.7.98

Dr. Peter Jarvis
Jussi Stader
University of Edinburgh, UK
9. - 10.7.98

Dipl. Inform. Peter Mohn
Institut für Energietechnik Halden, Norwegen
13.11.98

Prof. Dr. Frank Maurer
University of Calgary, Canada
10. - 23.12.98

1999

Prof. Dr. C. Mohan
INRIA Paris, Frankreich
4.2.99

Dr. Andreas Prinz
Humboldt-Universität Berlin
7. - 9.3.99

Dr. Uwe Glässer
Universität-GH Paderborn
7. - 9.3.99

Dipl. Ing. Christoph Nytsch
Institut für Elektrotechnik, TU Berlin
22.3.99

Dr. Khaled Ragab Abd-El-Tawab
Ain Shams University, Kairo / Ägypten
6. - 13.4.99

Dr. Jan de Meer
GMD Fokus Berlin
8. - 9.4.99

Dr. Jeff Magee
Imperial College, London / UK
7. - 8.6.99

Dipl. Inform. Serge Rielau
IBM Toronto / Canada
8.6.99

Dipl. Inform. Vladimir Tamara
Universidad de los Andes, Bogota / Columbien
10. - 17.6.99

Prof. Dr. Hartmut Wedekind
Universität Erlangen-Nürnberg
11.6.99

Dr. Sandrine Lampriere-Couffin
University Laboratory in Automated Production, Cedex/Frankreich
18. - 19.11.99

Prof. Dr. Julio Leite
PUC Rio de Janeiro, Brasilien
15.10.99 - 31.12.2000

2000

Prof. Dr. Julio Leite
PUC Rio de Janeiro, Brasilien
1.1.2000 - 29.2.2000

Dr. Joachim Thomas
UBS AG, Basel / Schweiz
13.1.2000

Dr. Martin Otter
Deutsches Zentrum für Luft- und Raumfahrt, Oberpfaffenhofen
20.1.2000

Dipl. Inform. Christoph Gollmick
Universität Jena
7.2.2000

Dipl. Inform. Mureed Hussain
Universität Marseille, Frankreich
8.2.2000

Prof. Carlton Pu
Georgia Institute of Technology, Atlanta, Georgia/USA
26.4.2000

Teilprojekt A1

Software Engineering Labor

Leiter

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

Mitarbeiter

Ergänzungsausstattung:

Dipl.-Inform. Raimund L. Feldmann
Dipl.-Ing. (FH) Thomas Schneider (ab 01.04.1999)
Dipl.-Ing. (FH) Wolfgang Scholz (bis 31.10.1998)
Dipl.-Inform. Stefan Vorwieger

Grundausstattung:

Dipl. Inform. Antje von Knethen

1 Kurzfassung der Aufgabenstellung

Das Teilprojekt A1 ist ein Infrastrukturprojekt, das die anderen Teilprojekte im Rahmen eines Software-Engineering(SE)-Labors bei der experimentellen Erprobung ihrer generischen Beschreibungstechniken und Entwicklungsmethoden unterstützt. Das Teilprojekt A1 übernimmt dabei die Rolle des Auftragnehmers für die Durchführung von Experimenten und Anwendungsentwicklungen.

1.1 Langfristige Aufgabenstellung

Das langfristige Ziel des SE-Labors ist es, Experimente in den Anwendungsgebieten des SFB effektiv zu unterstützen sowie die erzielten Ergebnisse und Erfahrungen für eine Wiederverwendung in späteren Experimenten[1] aufzubereiten und über eine Erfahrungsdatenbank zur Verfügung zu stellen.

1.2 Aufgabenstellung in der Förderperiode

In der ersten Förderperiode wurde ein initiales SE-Labor sowie eine prototypische Implementierung einer Erfahrungsdatenbank erstellt. In der zweiten Förderperiode sollte das SE-Labor als technologische Ausführungsplattform sowie die SFB-Erfahrungsdatenbank (SFB-EDB) als Ablage- und Wiederverwendungsplattform für Software-Engineering-Erfahrung soweit aufbereitet werden, dass die Durchführung und Ablage von Technologie-Experimenten und besonders von Baseline-Entwicklungen systematisch unterstützt werden kann. Die Aufgaben waren im einzelnen:

- Erweiterung des SE-Labors um notwendige kommerzielle Werkzeuge, besonders aber um SFB-interne Prototypen;
- Aufbau und Wartung der SFB-EDB, um experimentübergreifende Erfahrungen für die Wiederverwendung strukturiert ablegen zu können;
- Anleitung und Unterstützung bei der Durchführung von Experimenten;
- Ausdehnung der empirische Untersuchung der Effizienz des SE-Labors vom Werkzeugbereich auf die SFB-EDB.

2 Entwicklung des Teilprojekts in der Förderperiode

Die durch die Arbeitspakete gestellten Arbeiten wurden im Wesentlichen erfüllt. Im Bereich der SFB-EDB konnten große Fortschritte technologischer und inhaltlicher Art erzielt werden, die über die geplanten Arbeitspakete hinausgehen. Lediglich der

1. Unter dem Begriff *Experiment* werden im SFB folgende Arten von empirischen Studien verstanden: 1) *projekt-interne Technologieerprobungen*, welche der schrittweisen Erprobung von projekt-internen Ergebnissen dienen, 2) *SFB-öffentliche Technologie-Experimente* (kurz: *Technologie-Experimente*), um fertigentwickelte Beschreibungstechniken und Werkzeuge im Vergleich zu zuvor verwendeten Techniken bewerten zu können, sowie 3) *SFB-öffentliche Baseline-Entwicklungen* (kurz: *Baseline-Entwicklungen*), um im Rahmen einer kompletten Software-Entwicklung die Ergebnisse der Technologie-Experimente überprüfen zu können.

geplante Einsatz einer Prozessmaschine zur kontrollierten Durchführung von Experimenten (Arbeitspaket A3 u. C1) wurde in den Querschnittsprojekten durch personelle Kontrolle ersetzt. Die verfügbaren Prozessmaschinen, ProcessWeaver/CabWebFlow sowie MVP-S, boten nicht die notwendige Stabilität und Anpassbarkeit an die Bedürfnisse einer empirischen Software-Entwicklung, um Experimente ungefährdet durchführen zu können. Die in Teilprojekt A2 entwickelte Projektplanungs- und -ausführungsplattform MILOS, die ab der dritten Förderperiode zum Einsatz kommen soll, ermöglicht die Kontrolle mit dynamischer Anpassung des unterliegenden Entwicklungsprozesses sowie die empirische Begleitung von Software-Entwicklungen.

Die in diesem Berichtszeitraum zur Einreichung vorgesehene Dissertation von S. Vorwieger: *Umfassendes, planungsorientiertes Software-Konfigurationsmanagement für lernende Software-Organisationen* geht über das für das Arbeitspaket A2 geplante Vorhaben hinaus. Hier wird ein Ansatz zur Integration von Software-Konfigurationsmanagement(SKM)-Mechanismen in die prozessorientierte Projektplanung vorgestellt; ebenso behandelt die Arbeit die Verwaltung aller in einem Experiment anfallenden Ergebnisse zur verbesserten Unterstützung der Experiment-Analyse. Für die Planung und Ausführung wurde ein Werkzeug zur Verfügung gestellt.

3 Durchgeführte Arbeiten und Ergebnisse

3.1 Arbeitsschwerpunkt A: Erweiterung des SE-Labors

Das SE-Labor deckt zur Unterstützung der empirischen Software-Entwicklung drei Schwerpunkt-Technologien ab, deren Wartung und Erweiterung Ziel dieses Berichtszeitraumes waren: 1) *Plattform-Werkzeuge*, welche den Verbesserungsprozess unterstützen, wie z.B. Werkzeuge zur Erstellung von Projektplänen, Prozessmaschinen, Messwerkzeuge und Datenanalysewerkzeuge, 2) *Softwareentwicklungswerkzeuge*, welche die eigentliche Softwareentwicklung ermöglichen, wie z.B. Anforderungs-Beschreibungswerkzeuge, Testwerkzeuge, Compiler und 3) die *SFB-EDB* als Ablage

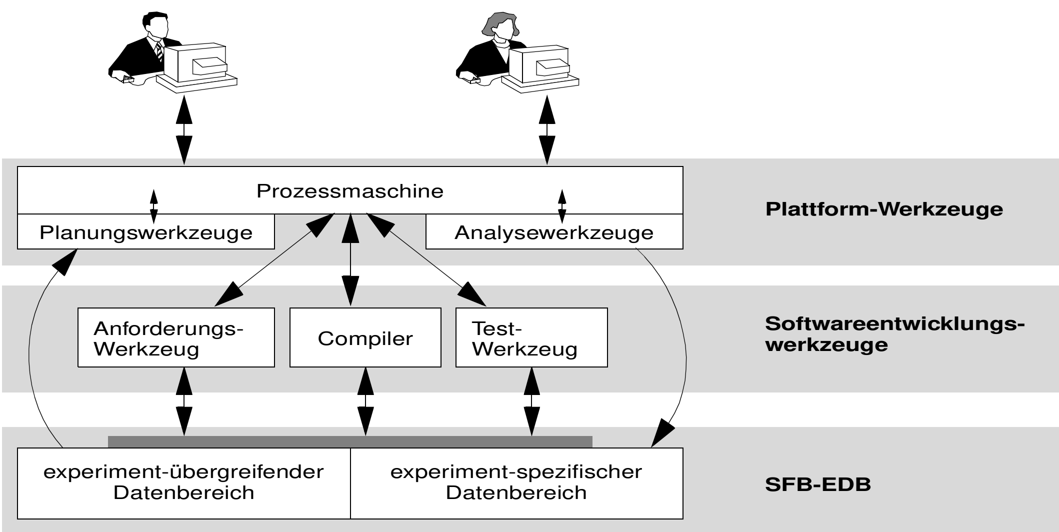

Abbildung 1: Schnittstellen zwischen den Technologie-Schwerpunkten des SE-Labors

und Wiederverwendungsplattform (s. Abb. 1). Die Ergebnisse zu Plattform- und Softwareentwicklungswerkzeugen werden im Folgenden erläutert; die Ergebnisse zur SFB-EDB finden sich im nachfolgenden Abschnitt 3.2

AP-A1: Werkzeuge für Software-Entwicklungstätigkeiten und Experimente

In der ersten Förderperiode lag der Fokus der in das SE-Labor zu integrierenden Plattform- und Softwareentwicklungswerkzeuge auf kommerziellen Werkzeugen. Diese wurden in der aktuellen Förderperiode kontinuierlich auf dem aktuellen Stand gehalten, z.B. für *SDT* von Telelogic, *Software-through-pictures(StP)* von Aonix oder der Java-Entwicklungsumgebung *jde* von Sun. Ergänzend wurden die in Tabelle 1 aufgeführten Werkzeuge —kommerzielle und SFB-Prototypen— in das SE-Labor integriert. Dieser Liste ist zu entnehmen, dass hauptsächlich Plattform-Werkzeuge in das SE-Labor eingebracht wurden. Die meisten dieser Werkzeuge wie bspw. MILOS oder PlaST^{pro} konnten bisher nicht im realen Experimentkontext eingesetzt werden, da sie sich derzeit in der Validierungsphase befinden.

Werkzeug/Komponente		Typ	TP	Beschreibung
GQM*	Aspect	2A†	B1	Textuelle Erstellung von GQM-Plänen
	Planner	2A		Generierung von GQM-Plänen
GEM*		2A	A1/B1	Graphische Modellierung von MVP-L Prozessmodellen
MoST*		2A	B1	Graphische Unterstützung für die Modellierung von MVP-L Prozessmodellen
ProTail		2A	B1	Generierung von Prozessmodellen
PlaST^{pro}	ProdOMo	2A	A1	Produktorientierte Modellierung von Entwicklungsprozessen [8]
	PlaST	2A		Prozessorientiertes SKM-Tool mit Anbindung an ProdOMo zur Verwaltung von SE-Erfahrungen [25]
MILOS	Modellierer	2A	A2	Modellierung von Prozessmodellen
	Planer	2A		Instanziierung von Prozessmodellen
	Process Engine	2A		Ausführung von Prozessmodellen
	Exporter	2A		Konvertierung von MS Project Plänen in MILOS
DictTool		2B	B5	Web-basiertes, datenbank-gestütztes Projekt-Dictionary
Xtelescreen		1A	-	Demo/Teaching-Tool für X-Windows Umgebungen

Tabelle 1: Liste der neu in das SE-Labor integrierten Werkzeuge
* Dieses Werkzeug liegt mit signifikant erweiterter Funktionalität vor
† (1): kommerzielles Werkzeug, (2): SFB-Prototy
(A): Plattform-Werkzeug, (B): SW-Entwicklungswerkzeug

AP-A2: Software-Konfigurationsmanagement (SKM)

Zur Unterstützung des Software-Konfigurationsmanagements sollte in dem Berichtszeitraum ein kommerzielles SKM-Werkzeug in das SE-Labor integriert werden. Hierzu wurden zwei Strategien parallel verfolgt:

Einerseits wurde eine Evaluierung von ca. 70 SKM-Werkzeugen vorgenommen und nach Kriterien gemäß der spezifischen Eigenschaften des SE-Labors bewertet. Neben einer Reihe von anderen Kriterien mussten die Werkzeuge an den SFB-Entwicklungsprozess, der laufenden Verbesserungen unterworfen ist, anpassbar sein. Des Weiteren musste eine einfache Möglichkeit zur Anpassung von Schnittstellen für die SFB-EDB und die Prozessmaschine MILOS gewährleistet sein. Auch sollte das Werkzeug im Hinblick auf eine post-mortem-Analyse in der Lage sein, Prozesse und Messdaten entsprechend zu verwalten und in Bezug zu den übrigen Entwicklungsprodukten zu setzen [45]. Im finanziell sinnvollen Rahmen gab es kein Werkzeug, welches diesen Anforderungen genügte.

Andererseits wurde parallel dazu die Möglichkeit zur Entwicklung eines SFB-eigenen Prototypen untersucht. Hierzu wurden zuverlässige Werkzeuge zur Versionierung mit einfachen Möglichkeiten zur Erweiterung als Kern in Betracht gezogen: RCS, RCE als kommerzieller Nachfolger von RCS sowie CVS. CVS und RCE wurden jedoch nach eingehender Prüfung ausgeschlossen. Ein erster Prototyp zur prozess-abhängigen Verwaltung von SW-Artefakten und Konfigurationen mit RCS als Versionierungskern wurde in [19] erfolgreich erprobt. Da jedoch bei der Weiterentwicklung festgestellt wurde, dass die technischen Beschränkungen durch RCS zu groß wurden, wurde ein komplett eigener Prototyp —PlaST^{pro}— entwickelt [8][25].

AP-A3: Integration von Softwareentwicklungswerkzeugen in Prozessmaschinen

Dieses Arbeitspaket sah vor, die SFB-prototypischen Prozessmaschinen MVP-S und MILOS im SE-Labor bereitzustellen und für beide in Zusammenarbeit mit den Teilprojekten A2, A3 und B1 eine einheitliche Werkzeugschnittstelle zu definieren. Da beide Werkzeuge auf den Konzepten der Prozessmodellierungssprache MVP-L basieren, jedoch gänzlich unterschiedliche Schnittstellen anbieten, wurde die Entscheidung getroffen, das modernere MILOS des Teilprojekts A2 in den Mittelpunkt zu stellen. Für MILOS wurden Schnittstellen für kommerzielle Werkzeuge definiert und realisiert. Es ist vorgesehen, MILOS ab der folgenden Förderungsperiode für Experimente einzusetzen (s. Teilprojekt A2).

3.2 Arbeitsschwerpunkt B: Aufbau und Wartung einer experiment-übergreifenden Erfahrungsdatenbank

Ziel dieses Arbeitsschwerpunktes war die Definition und initiale Implementierung einer SFB-EDB zur Verwaltung des im SFB 501 anfallenden experiment-übergreifenden Erfahrungswissens. Die Definition des logischen Integrationsschemas der SFB-EDB in den experimentellen Erprobungs- und Lernzyklus des SFB 501 erfolgte dabei in Kooperation mit dem Teilprojekt B1. Die Ergebnisse sind in [34] dokumentiert.

Zunächst wurde aus Gründen einer schnelleren Verfügbarkeit und eines sich noch in der Entwicklung befindlichen Ablageschemas ein web-basierter Prototyp der SFB-EDB erstellt [33]. Dieser wurde von A1 entwickelt und zentral im SE-Labor implementiert, von wo aus er über das Intranet des SFB 501 für alle Teilprojekte zugreifbar ist [13]. Anhand dieses Prototyps wurde systematisch das Schema für die experimentübergreifenden Teile der SFB-EDB entwickelt [42][44][52]. Dabei wurde von Beginn an großer Wert auf eine enge Zusammenarbeit mit Wissenschaftlern anderer Forschungseinrichtungen, wie zum Beispiel dem Fraunhofer IESE, gelegt. Daraus resultiert u. A. der Ansatz zur Intergration von Technologiebeschreibungen in das derzeitige Schema der SFB-EDB [18] [36].

Basierend auf dem mit Hilfe des web-basierten Prototypen erprobten Schemas für das experiment-übergreifende Erfahrungswissen in der SFB-EDB wurden die Anforderungen für eine durch ein Datenbanksystem unterstützte Reimplementierung der SFB-EDB aufgestellt. Mit Hilfe dieser Informationen wurde durch das Teilprojekt A3 ein objektrelationales Datenbanksystem als geeignete Technologie für die Umsetzung ausgewählt [41].

Aus diesem Arbeitsschwerpunkt gingen zudem einige weiterführende Arbeiten hervor, die ursprünglich nicht Bestandteil des Finanzierungsantrags für die Förderperiode 1998 - 2000 waren:

- Erfahrungen mit dem web-basierten Prototyp der SFB-EDB verdeutlichten die Notwendigkeit für ein verbessertes Zugriffskonzept auf die Inhalte der SFB-EDB. Die daraus resultierenden Überlegungen schlugen sich in der Anforderungsbeschreibung für die Reimplementierung der SFB-EDB nieder [17]. Als Lösung wurde ein Rollenkonzept für Erfahrungsdatenbanken entwickelt [24] [49]. Mit diesem Konzept können Zugriffsrechte sowie geeignete Benutzungsschnittstellen gezielt verwaltet und zugeteilt werden.
- Des Weiteren wurde ein geeignetes Kommunikationsprotokoll zur Übermittlung von Daten zwischen dem Server der Erfahrungsdatenbank und den darauf zugreifenden Clients entwickelt [10]. Dies geschah im Rahmen einer internationalen Zusammenarbeit mit dem Fraunhofer Center Maryland (USA) und dem Fraunhofer IESE in Kaiserslautern.

Diese zusätzlichen Arbeiten dienen somit u. A. als Grundlage für die geplante Weiterentwicklung der SFB-EDB in der neu beantragten dritten Förderperiode. Nachfolgend werden die konkreten Ergebnisse der einzelnen Arbeitspakete beschrieben:

AP-B1: Integration der experiment-spezifischen Datenbereiche

In diesem Arbeitspaket sollten möglichst rasch die einzelnen Experimentergebnisse zusammengefasst und in die SFB-EDB integriert werden. Die ursprüngliche Planung sah zunächst eine Realisierung auf der Basis eines objekt-orientierten Datenbanksystems vor. Diese Lösung stellte sich jedoch als zu aufwendig und daher nicht zweckdienlich heraus und wurde folglich verworfen. Als Alternative wurden die experiment-spezifischen Datenbereiche mit den experiment-übergreifenden Datenbereichen in einem web-basierten Prototypen der SFB-EDB zusammengefasst [13] [33]. Abbildung 2 illustriert, wie die experiment-spezifischen und -übergreifenden Datenbereiche der web-basierten SFB-EDB zusammenhängen. Ein Vorteil dieser Lösung ist, dass alle bisher verwendeten Werkzeuge —Eigenentwicklungen gleichermassen wie Kommerzielle— weiterhin über das Dateisystem auf die zu bearbeitenden Daten

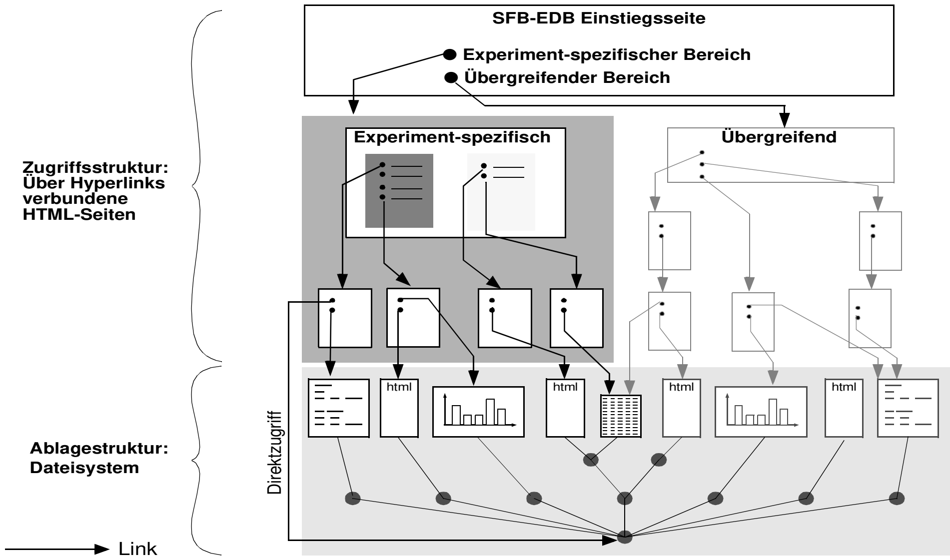

Abbildung 2: Ablage- und Zugriffstruktur des web-basierten SFB-EDB Prototyps

zugreifen können. Eine spezielle Anpassung der Werkzeuge für den Zugriff auf die experiment-spezifischen Datenbereiche während einer Experimentdurchführung entfiel damit in dieser frühen Entwurfsphase der SFB-EDB.

Für die einzelnen Typen von Fallstudien wurden geeignete Ablage- und Zugriffsstrukturen entwickelt [1]. Diese erlauben eine systematische und fortlaufende Integration von Experiment-Ergebnissen in die SFB-EDB. Basierend auf diesen Überlegungen wurden entsprechende Werkzeuge zur (teil-)automatischen Erstellung dieser Ablage- und Zugriffsstrukturen im web-basierten Prototypen der SFB-EDB entwickelt [23].

AP-B2: Realisierung der Kontextsuche

Ziel dieses Arbeitspakets war es, die Browsing-Schnittstelle der SFB-EDB für den Direkt- und Beziehungszugriff um eine weitere Funktionalität zu erweitern. Hierbei ist es möglich, basierend auf einer stichwortartigen Beschreibung eines konkreten Kontextes, auf Erfahrungen in der SFB-EDB zu zu greifen, die diesem Kontext entsprechen. Zur Beschreibung des Kontexts wurde für alle Erfahrungselemente in den einzelnen Bereichen der SFB-EDB ein Meta-Schema entwickelt [44][52]. Die darin enthaltenen Informationen werden zur Suche heran gezogen. Dabei sollte die Suche nach Möglichkeit ähnlichkeitsbasiert erfolgen, um in die Treffermenge auch geeignete, ähnliche Elemente mit ein zu beziehen. Im web-basierten Prototyp der SFB-EDB war eine derartige Realisierung der Kontextsuche mit vertretbarem Aufwand technisch nur bedingt möglich. Insbesondere die Möglichkeiten zur ähnlichkeitsbasierten Suche wurden erst mit der Einführung der objektrelationalen Datenbanktechnologie möglich [41].

AP-B3: Wartung von Erfahrungen

Ziel dieses Arbeitspakets war es, eine schritthaltende Aufbereitung und Integration von Ergebnissen aus aktuellen Experimenten in die SFB-EDB zu gewährleisten. Als Basis hierfür diente die im Arbeitspaket B1 entwickelte Ablage- und Zugriffsstruktur für Fallstudien. Im Berichtszeitraum wurden mehrere Experimente durchgeführt. Als Baseline-Entwicklungen wurden die beiden Querschnittsprojekte Q1 und Q2 [19] [26] [54] in die SFB-EDB integriert. Des Weiteren wurde der im Teilprojekt B4 entwickelte SDL-Pattern-Ansatz als initiales Beispiel komplett in die SFB-EDB eingebunden [16] [35]. Neben den eigentlichen SDL-Patterns wurden eine Sammlung von Fragebögen zur quantitativen Evaluierung der SDL-Patterns, ein SDL-Glossar sowie die Prozessbeschreibung des gesamten Ansatzes in den experiment-übergreifenden Teil der SFB-EDB eingebracht [15] [37]. Die aus einer ersten Fallstudie gewonnenen Erkenntnisse wurden genutzt, um die SDL-Pattern-Sammlung in der SFB-EDB zu ergänzen und das verwendete initiale Messprogramm zu verbessern [20] [43].

3.3 Arbeitsschwerpunkt C: Anleitung und Unterstützung bei der Durchführung von Experimenten

Dieser Arbeitsschwerpunkt betrifft die technologische und personelle Unterstützung bei der Planung, Durchführung und Analyse von Experimenten durch das SE-Labor. Ziel war es, die technologische Basis insofern zu verbessern, dass der personelle Aufwand bei diesen Tätigkeiten gerade für das Personal von Teilprojekt A1 reduziert werden kann. Des Weiteren sollten die automatisierbaren Aktivitäten des experimentellen Lernprozesses in einer Prozessmaschine realisiert werden.

AP-C1: Realisierung einer Prozessmaschine für Experimentdurchführung

In diesem Arbeitspaket sollte die in Teilprojekt B1 entwickelte Experimentiermethodik mittels einer Prozessmaschine umgesetzt werden, um eine genauere Kontrolle über den Experimentablauf zu erhalten und den personellen Aufwand für Betreuung zu reduzieren. Hierzu wurde zunächst die kommerzielle Prozesskontroll-Umgebung Process*WEAVER* und sein Nachfolger CabWebFlow in Betracht gezogen. Es stellte sich jedoch schnell heraus, dass beide Werkzeuge, die auf einer identischen Ausführungsmaschine beruhen, in einem empirischen Umfeld nicht geeignet sind. Die Integration der Messdatenerfassung und eine von den Messwerten gesteuerte Durchführung des Experiments (interner Feedback-Loop) ist technisch nur äußerst aufwendig und kaum nachvollziehbar, d.h. nicht wartbar, zu realisieren. Außerdem bieten beide Werkzeuge keine Möglichkeit zur dynamischen Anpassung der Entwicklungsprozesse während der Durchführung. Ein starres Prozessmodell ist nur für lang erprobte, identisch wiederholbare Entwicklungen —wie sie im SFB naturgemäß *nicht* vorkommen— sinnvoll. Es wurde daher die Entscheidung getroffen, die Integration von MILOS in das SE-Labor abzuwarten, da MILOS bei den genannten Schwachstellen Verbesserungen verspricht. Alle Experimente im Berichtszeitraum wurden zwar werkzeug-gestützt geplant und analysiert, jedoch rein personell in der Durchführungsphase betreut.

AP-C2: Anleitung bei der Durchführung von Experimenten

Im Berichtszeitraum wurden eine Reihe von Experimenten, darunter zwei Baseline-Entwicklungen [19] [26] [54] sowie weitere Technologie-Experimente (z.B. [15]) mit Unterstützung des A1-Personals durchgeführt. Die Unterstützung lag dabei schwerpunktmäßig auf der Planung, der Messdatenerfassung und Produktverwaltung sowie der Analyse der erfassten Messdaten und qualitativen Erfahrung. Technologische Verbesserungen gab es im experiment-spezifischen und -übergreifenden Datenbereich der SFB-EDB: Die bisher prototypisch vorhandenen Zugriffs- und Ablagestrukturen für den experiment-spezifischen Datenbereich der SFB-EDB wurden evaluiert und verbessert [23]. Für einen vereinfachten Zugriff auf die SFB-EDB wurden ein Werkzeug zur Erzeugung einer initialen Ablage- und Zugriffsstruktur sowie die Kontextsuche auf Basis des objektrelationalen Datenbanksystems (s. Arbeitspaket B2) zur Verfügung gestellt.

3.4 Arbeitsschwerpunkt D: Empirische Untersuchung der Effizienz des SE-Labors

Ziel dieses Arbeitsschwerpunktes ist es, die Nutzung des SE-Labors und der in ihm installierten Werkzeuge sowie der SFB-EDB zu bewerten. Die bisherigen Ergebnisse des dafür vorgesehenen Arbeitspakets werden nachfolgend kurz beschrieben:

AP-D1: Evaluation der Effizienz des SE-Labors

Zur kontinuierlichen Untersuchung der Benutzungszugriffe auf die Werkzeuge des SE-Labors wurde zunächst das seit März 1997 existierende System zur Protokollierung der Werkzeugaufrufe um die Protokollierung der Benutzung des web-basierten Prototypen der SFB-EDB erweitert. Insbesondere wurde hierzu ein Mechanismus entwickelt, der die spezifische Benutzung aufgeschlüsselt nach Inhalten der SFB-EDB protokolliert. Dazu wurden die Log-Dateien des Web-Servers, auf dem die SFB-EDB läuft, zur Auswertung herangezogen. Zugleich wurde dem System eine teilautomatisierte, rudimentäre grafische Aufbereitung der erhobenen Messdaten hinzugefügt. Abbildung 3 zeigt eine solche teilautomatisch erstellte Auswertung. Mit Hilfe der so gewonnenen Auswertungen konnten wichtige Rückschlüsse auf die Akzeptanz der SFB-EDB gezogen werden. Es konnten so Hinweise bezüglich der Entwicklung des

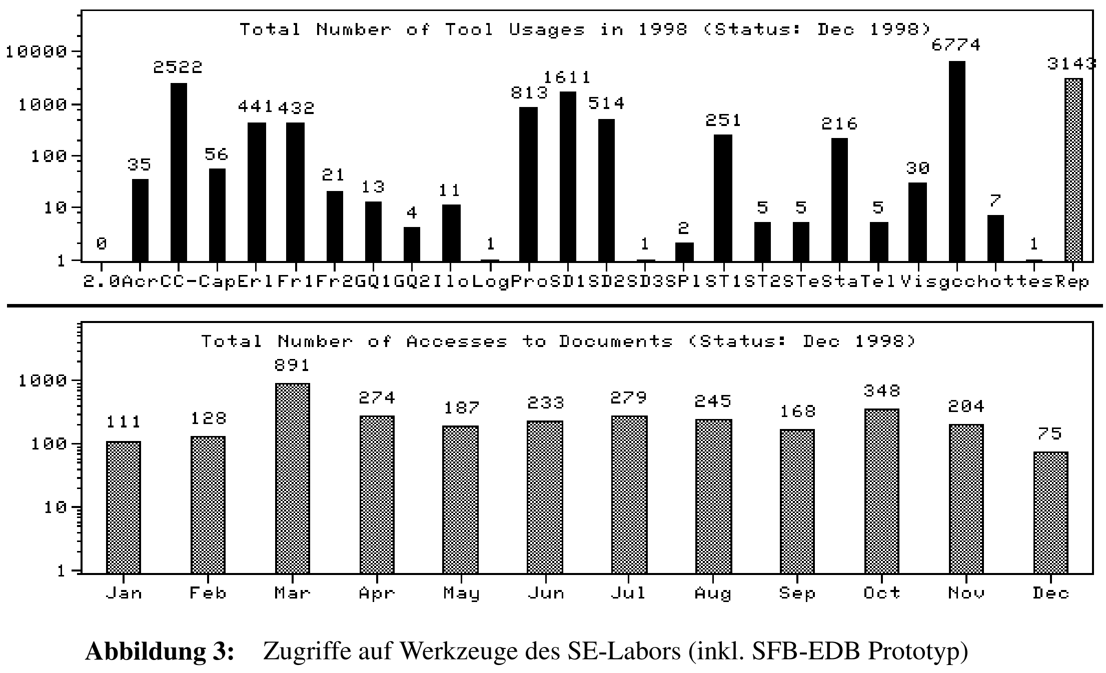

Abbildung 3: Zugriffe auf Werkzeuge des SE-Labors (inkl. SFB-EDB Prototyp)

SFB-EDB-Schemas gewonnen werden. Eine abschließende Bewertung aller erhobenen Daten zur Werkzeugnutzung des SE-Labors steht zum gegenwärtigen Zeitpunkt noch aus, ist aber zum Ende der Förderperiode vorgesehen.

Um die wichtigen Nutzungsinformationen bezüglich der Zugriffe auf Inhalte der SFB-EDB auch nach der Umstellung auf ein objektrelationales Datenbanksystem zu erhalten, wurden erste Schritte zur Definition eines verbesserten, und durch die bessere Technologieunterstützung detaillierteren Messprogramms für die SFB-EDB unternommen [7]. Das Messprogramm wurde in enger Zusammenarbeit mit dem Fraunhofer IESE und dem Fraunhofer Center Maryland definiert [48].

4 Zusammenarbeit mit anderen Teilprojekten

Aufgrund seines Dienstleistungscharakters steht das SE-Labor mit allen anderen Teilprojekten in engem Kontakt. Im Folgenden werden daher nur die außerhalb dieses Rahmens liegenden Kooperationen beschrieben:

In enger Kooperation mit dem Teilprojekt B1 wurde die Definition des logischen Integrationsschemas der SFB-EDB in den experimentellen Erprobungs- und Lernzyklus des SFB 501 entwickelt. Die Ergebnisse sind u. A. in [34] dokumentiert und wurden einem breiten Fachpublikum durch einen Vortrag auf einer internationalen Konferenz präsentiert.

In Kooperation mit dem Teilprojekt B4 wurde eine vollständige Integration des dort entwickelten SDL-Pattern-Ansatzes in die SFB-EDB vorgenommen [46]. Zugleich wurde ein maßgeschneidertes, auf der GQM-Methodik aufbauendes Messprogramm zur Evaluierung und kontinuierlichen Verbesserung des SDL-Pattern-Pools sowie des gesamten Ansatzes entwickelt [15][16]. Dieses wurde anhand einer ersten Fallstudie [53], die von A1 methodisch mit betreut wurde, evaluiert [20] und in den SDL-Pattern-Ansatz in Form eines kontinuierlichen Lernprozesses integriert [37][43]. Zugleich dient der im Teilprojekt B4 entwickelte SDL-Pattern-Ansatz als initiales Anwendungsbeispiel für die web-basierte Prototypimplementierung der SFB-EDB innerhalb des SFB 501 [35].

Die Weiterentwicklung der SFB-EDB erfolgte in enger Kooperation mit dem Teilprojekt A3. Hierbei wurde die Basis zu einer, auf objektrelationaler Datenbanktechnologie beruhenden, verbesserten Implementierung der SFB-EDB gelegt. Diese bildet auch eine wichtige Grundlage für die geplanten Aktivitäten in der neu beantragten Förderperiode. Zahlreiche interne und externe Veröffentlichungen [17][41][50] sowie Vorträge und Tool-Demonstrationen [28] auf nationalen und internationalen Veranstaltungen dokumentieren diese gute Zusammenarbeit. Dabei diente erneut, in Kooperation mit dem Teilprojekt B4, der SDL-Pattern-Ansatz als initiales Anwendungsbeispiel bei der Erprobung der neuen Technologie [22][50].

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] M. Fechtig: *Bewertung der Zugriffs- und Ablagestruktur für Fallstudien der SFB-Erfahrungsdatenbank mittels Integration von SE-Praktika*; FB Informatik, Universität Kaiserslautern, Jan. 1998.

[2] M. Habetz: *Werkzeuge zur Unterstützung der Experience Factory des Sonderforschungsbereichs 501*; FB Informatik, Universität Kaiserslautern, Dez. 1998.

[3] T. Willrich: *Erstellung eines grafischen Frontends zur Verwaltung von System-Konfigurationen für SE-Fallstudien*; FB Informatik, Universität Kaiserslautern, Dez. 1998.

[4] L. Achterfeldt: *Ein Modul zur projekt-spezifischen Konfigurierung des Produktverwaltungssystems PlaST*; FB Informatik, Universität Kaiserslautern, 2000.

[5] H. Damczyk: *Rollenorientierte Client-Server-Umgebung für das Produktverwaltungssystem PlaST*; FB Informatik, Universität Kaiserslautern, 2000.

[6] J. Mentges: *Anpassbare Definition und analyseorientierte Erfassung von Meßdaten als Modul des Produktverwaltungssystems PlaST*; FB Informatik, Universität Kaiserslautern, 2000.

[7] M. Frey: *Measurement Based Evaluation of a Running Experience Base*; FB Informatik, Universität Kaiserslautern, Mai 2000.

Diplomarbeiten

[8] O. Swienty: *Ein Ansatz zur Integration von Mechanismen des Produktmanagements in die Prozeßmodellierung*; FB Informatik, Universität Kaiserslautern, Jan. 2000.

[9] M. Fechtig: *Definition und Visualisierung von Projekthistorien und Projektzuständen als Modul eines Produktverwaltungssystems*; FB Informatik, Universität Kaiserslautern, 2000.

[10] M. Habetz: *Design and Evaluation of a Role-Based Communication Protocol for Distributed Access to Experience Bases*; FB Informatik, Universität Kaiserslautern, April 2000.

[11] J. Rech: *Integration und Repräsentation von existierenden Erfahrungen und Datenbeständen in einer Erfahrungsdatenbank*; FB Informatik, Universität Kaiserslautern, Aug. 2000.

Dissertationen

[12] S. Vorwieger: *Umfassendes planungsorientiertes Software-Konfigurationsmanagement für lernende Software-Organisationen*; (für den Berichtsraum zur Einreichung vorgesehene) Dissertation, Universität Kaiserslautern.

SFB-Berichte

[13] R. L. Feldmann, S. Vorwieger: *The web-based Interface to the SFB 501 Experience Base*; SFB Report 01/98, FB Informatik, Universität Kaiserslauter, 1998.

[14] M. Habetz: *Demo-Viewer – Ein Frontend zur Präsentation der Teilprojekte des SFB 501*; SFB Report 02/98, FB Informatik, Universität Kaiserslautern, 1998.

[15] R. L. Feldmann, B. Geppert, F. Rößler: *Towards an Experimental Evaluation of SDL-Pattern based Protocol Design*; SFB Report 04/98, FB Informatik, Universität Kaiserslautern, 1998.

[16] R. L. Feldmann, B. Geppert, F. Rößler: *A Repository-Based Improvement Cycle for Reuse-Driven SDL* System Development; SFB Report 11/98, FB Informatik, Universität Kaiserslautern, 1998.

[17] R. L. Feldmann, W: Mahnke, N: Ritter: *(OR)DBMS-Support for the SFB 501 Experience Base*; SFB Report 12/98, FB Informatik, Universität Kaiserslautern, 1998. (in Bearbeitung).

[18] R. L. Feldmann, C. Tautz: *Improving Best Practices Through Explicit Documentation of Experience About Software Technologies*; Report 053.98/E, Fraunhofer IESE, Kaiserslautern, October 1998.

[19] R. L. Feldmann, J. Münch, S. Queins, S. Vorwieger, G. Zimmermann: *Baselining a Domain-Specific Software Development Process;* SFB Report 02/99, FB Informatik, Universität Kaiserslautern, 1999.

[20] R. L. Feldmann, B. Geppert, F. Rößler: *First Results from an Experimental Evaluation of SDL-Pattern based Protocol Design;* SFB Report 03/99, FB Informatik, Universität Kaiserslautern, 1999.

[21] M. Habetz: *Tool Support for the Software Engineering Laboratory*; SFB Report 04/99, FB Informatik, Universität Kaiserslautern, 1999.

[22] R. L. Feldmann, B. Geppert, W. Mahnke, N. Ritter, F. Rößler: *The ORDBMS-based SFB 501 Experience Base –Exemplified by the SDL-Pattern Approach–*; SFB Report 08/99, FB Informatik, Universität Kaiserslautern, 1999.

[23] S. Vorwieger, H. Damczyk, M. Fechtig, B. Schadt, O. Swienty: *Struktur und Werkzeuge des experimentspezifischen Datenbereichs der SFB 501 Erfah-*

rungsdatenbank; SFB Report 11/99, FB Informatik, Universität Kaiserslautern, 1999.

[24] R. L. Feldmann, M. Frey, M. Habetz: *Applying Roles to the SFB 501 Experience Base*; SFB Report 02/00, FB Informatik, Universität Kaiserslautern, 2000.

[25] S. Vorwieger: *PlaST- Ein Client-Server System zur prozeßorientierten umfassenden Produktverwaltung*; SFB Report XX/00, FB Informatik, Universität Kaiserslautern, 2000. (in Bearbeitung).

[26] A. von Knethen, R. L. Feldmann, J. Münch, S. Vorwieger: *A Baseline for the Development of Embedded Software*; SFB Report 04/00, FB Informatik, Universität Kaiserslautern, 2000. (in Bearbeitung).

Vorträge

[27] H. D. Rombach: Software Experience Factory: Basis für beschleunigte Prozeßverbesserung; GI-Workshop, Kaiserslautern, April 1999.

[28] R. L. Feldmann: The SFB 501 Experience Base; Experience Factory technology workshop, IESE Kaiserslautern. 16. Juli 1998.

[29] H. D. Rombach: Continuous Improvement of Software Development Competence: Prerequisite for Future Competitiveness; University of New South Wales, New South Wales, Australia, Nov. 1999.

[30] H. D. Rombach: Experimental Software Engineering; Colloquium, Nara Institute of Science and Technology (NAIST), School of Information Science, Nara, Japan, Nov. 1999.

[31] H. D. Rombach: Capitalizing on Experience; Industry Seminar, Software Engineering Association Australia (SEA), Sydney, Australia, November 1999.

[32] H. D. Rombach: Software Experience Factory: “How to Do It Guide”; SEA, Melbourne, Australia, Nov. 1999.

Tagungsbeiträge

[33] R. L. Feldmann, S. Vorwieger: *Providing an Experience Base in a research Context via the Internet*; in Proc. of the ICSE 98 Workshop on "Software Engineering over the Internet", http://sern.cpsc.ucalgary.ca/maurer/ICSE98WS/ICSE98WS.html, April 1998.

[34] R. L. Feldmann, J. Münch, S. Vorwieger: *Towards Goal-Oriented Organizational Learning: Representing and Maintaining Knowledge in an Experience Base*; in Proc. of the 10th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE‘98), San Francisco Bay, CA, USA, June 1998.

[35] B. Geppert, F. Rößler, R. L. Feldmann, S. Vorwieger: *Combining SDL Patterns with Continuous Quality Improvement: An Experience Factory Tailored to SDL Patterns*; in Proc. of the 1st Workshop of the SDL Forum Society on SDL and MSC (SAM98), Berlin, 1998. ISSN: 0863-095.

[36] R. L. Feldmann, C. Tautz: *Improving Best Practices Through Explicit Documentation of Experience About Software Technologies*; in C. Hawkins, M. Ross, G. Staples, J. B. Thompson (eds.), INSPIRE III Process Improvement Through Training and Education, British Computer Society, Sep. 1998.

[37] B. Geppert, F. Rößler, R. L. Feldmann: *Systematic Evolution of Reusable SDL Patterns*; in Proc. of the European Reuse Workshop, Madrid, Spain, Nov. 1998.

[38] A. von Knethen, E. Kamsties, R. Reussner, B. Shen: *A Comparative Case Study with Industrial Requirements Engineering Methods*; 11th Int. Conf. on Software Engineering and its Applications, Paris, France, Dec. 1998.

[39] E. Kamsties, A. von Knethen, J. Philipps, B. Schätz: *Eine vergleichende Fallstudie mit CASE-Werkzeugen für objektorientierte und funktionale Modellierungstechniken*; OMER Workshop, P. Hofmann, A. Schürr (eds.), OMER Workshop Proc., Bericht Nr. 1999-01, Mai 1999.

[40] E. Kamsties, A. von Knethen, J. Philipps, B. Schätz: *Eine vergleichende Fallstudie mit CASE-Werkzeugen für formale und semi-formale Beschreibungstechniken*; GI/ITG-Fachgespräch formale Beschreibungstechniken für verteilte Systeme, 1999.

[41] N. Ritter, H.-P. Steiert, W. Mahnke, R. L. Feldmann: *An Object-Relational SE-Repository with Generated Services*; in Proc. of the 1999 Information Resources Management Association Int. Conf., Hershey, PN, USA, May 1999.

[42] R. L. Feldmann: *Developing a Tailored Reuse Repository Structure –Experience and First Results–*; in Proc. of the Workshop on Learning Software Organizations (LSO‘99), Kaiserslautern, June 1999.

[43] R. L. Feldmann, B. Geppert, F. Rößler: *Continuous Improvement of Reuse–Driven SDL System Development*; in Proc. of the 11th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE‘99), Kaiserslautern, June 1999.

[44] R. L. Feldmann: *Developing a Repository Structure that Allows Comprehensive Reuse and Continuous Improvement*; in Proc. of the 8. Kolloquium Software-Entwicklung, Ostfildern, Technische Akademie Esslingen, Sep. 1999.

[45] S. Vorwieger: *Systematische Integration von Prozeß- und Produktmanagement*; GI'99, Jahrestagung der Gesellschaft für Informatik 1999, Springer Tagungsband, 5-9, Okt. 1999.

[46] R. L. Feldmann, B. Geppert, F. Rößler: *An Integrating Approach for Developing Distributed Software Systems -Combining Formal Methods, Software Reuse, and the Experience Base-*; in Proc. of the 5th IEEE Int. Conf. on Engineering of Complex Computer Systems, Las Vegas, Nevada, USA, Okt. 1999.

[47] H. D. Rombach: Experimentation: *Engine for Applied Research and Technology Transfer in Software Engineering*; in Proc. of the 24th annual Software Engineering Workshop (SWE24), Greenbelt, Maryland, USA, Dec. 1999.

[48] R. L. Feldmann, M. Nick, M. Frey: *Towards Industrial-Strength Measurement Programs for Reuse and Experience Repository Systems*; in Proc. of the 2nd Int. Workshop on Learning Software Organizations (LSO'2000), Oulu, Finnland, June 2000.

[49] R. L. Feldmann, M. Frey, M. Habetz, M. Mendonça: *Applying Roles in Reuse Repositories*; in Proc. of the 12th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE'2000), Chicago, IL, USA, July 2000.

[50] R. L. Feldmann, B. Geppert, W. Mahnke, N. Ritter, F. Rößler: *An ORDBMS-based Reuse Repository Supporting the Quality Improvement Paradigm – Exemplified by the SDL-Pattern Approach–*; in Proc. of the TOOLS USA 2000 34th Int. Conf. & Exhibition, Santa Barbara, CA, USA, Aug. 2000.

Zeitschriften / Monografien

[51] J. Avenhaus, R. Gotzhein, T. Härder, L. Litz, K. Madlener, J. Nehmer, M. Richter, N. Ritter, H. D. Rombach, B. Schürmann, G. Zimmermann: *Entwicklung großer Systeme mit generischen Methoden - Eine Übersicht über den Sonderforschungsbereich 501*. Informatik, Forschung und Entwicklung, 13(4):227–234, Dez. 1998.

[52] R. L. Feldmann: *On Developing a Repository Structure Tailored for Reuse with Improvement*; Accepted for publishing in: F. Bomarius, G. Ruhe (eds.), Learning Software Organization – Methodology and Applications. Lecture Notes in Computer Science Nr. 1756, Springer–Verlag, 2000.

Fremdveröffentlichungen

[53] S. van Lier: *SDL-Pattern basiertes Reengineering von RTP (Realtime Transport Protocol)*; FB Informatik, Universität Kaiserslautern, Aug. 1998.

[54] S. Queins, G. Zimmermann: *A First Iteration of a Reuse-Driven, Domain-Specific System Requirements Analysis Process*; SFB Report 13/99, FB, Universität Kaiserslautern, 1999.

Teilprojekt A2

Entwicklung einer flexiblen Modellierungs- und Ausführungsumgebung für Software-Entwicklungsprozesse

Leiter

Prof. Dr. Michael Richter
geb. 21.6.1938
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-2800
Fax: 0631/205-3557
richter@informatik.uni-kl.de

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

Mitarbeiter

Ergänzungsausstattung:

Dipl-Inform. S. Goldmann (bis 30.8.99)
Dipl-Inform. E. Petersen (seit 1.9.99)

Grundausstattung:

Dipl-Inform. B. Kötting

1 Kurzfassung der Aufgabenstellung

Ziel dieses Teilprojekts war die Entwicklung eines Werkzeugs zur integrierten Unterstützung von management- und technisch-orientierten Prozessen im Rahmen von Software-Entwicklungsprojekten sowie dessen experimentelle Erprobung. Ein Schwerpunkt lag dabei auf der Verzahnung von Planung und Ausführung. Es sollte ermöglicht werden, auch während der Ausführung eines Projekts den zugrundeliegenden Projektplan zu ändern bzw. zu verfeinern, was die Erfassung und Verwaltung kausaler Abhängigkeiten des Plans erfordert. Dies ist notwendig, um einerseits wieder zu einem konsistenten Projektzustand zu gelangen und andererseits eine automatische Benachrichtigung betroffener Mitarbeiter zu ermöglichen. Das resultierende Werkzeug sollte seine Nützlichkeit anhand von Experimenten demonstrieren.

2 Entwicklung des Teilprojekts in der Förderperiode

Das im Rahmen dieses Teilprojekts entwickelte MILOS-System integriert den in Teilprojekt B1 entwickelten MVP-Ansatz mit dem in Teilprojekt B2 entwickelten CoMo-Kit-Ansatz auf der Basis einer gemeinsamen Prozessmodellierungssprache. Der Prototyp wurde in der Programmiersprache Java in Kooperation mit der Arbeitsgruppe von Prof. Maurer (assoz. Mitglied des SFB, Universität Calgary, Kanada) entwickelt und basiert auf einer „three-tiered“-Architektur (siehe Abb. 1). Die Clients sind als Java Applets ausgelegt, um einerseits den Installationsaufwand gering zu halten und andererseits die Verwendbarkeit unter heterogenen Betriebssystemen zu gewährleisten, sofern für diese ein Java-kompatibler Browser existiert.

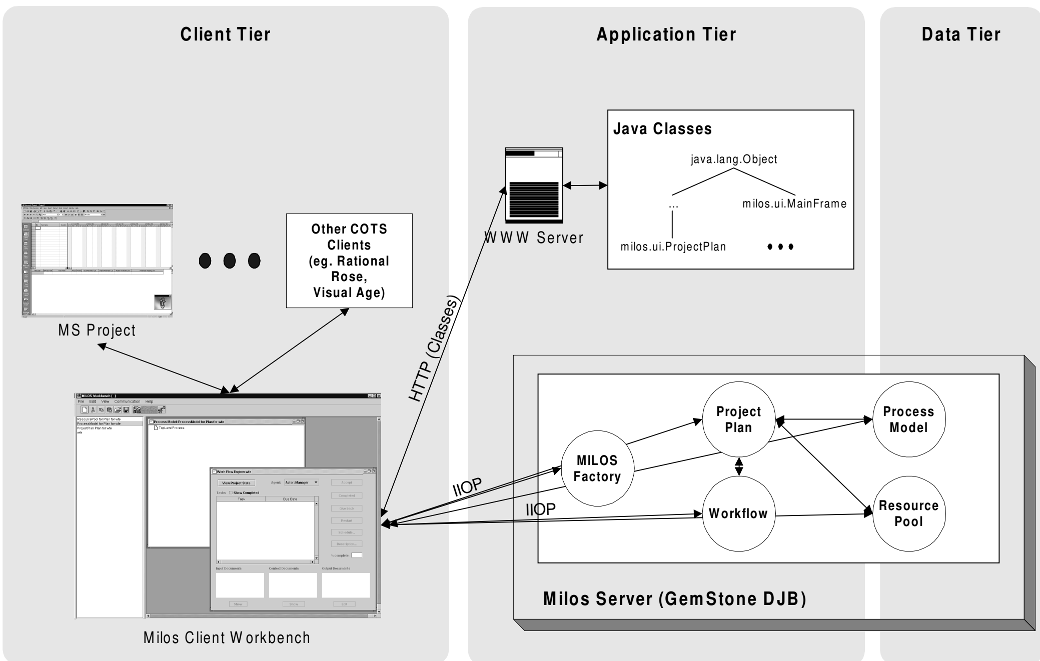

Abbildung 1: MILOS - GemStone Applikationsmodell

Die Serverkomponenten sind in das objektorientierte DBMS GemStone integriert, welches für Persistenz- und Transaktionsdienste verantwortlich ist.

Im Laufe des Teilprojekts wurden der WETICE-Workshops "Coordinating Distributed Software Development Projects" 1998 [27] und 1999 [29] sowie der ICSE-Workshops "Internet-based Software Development" 1999 [31] und 2000 in Zusammenarbeit mit Prof. Maurer organisiert. Hier und auf der SEKE 99 [23] [33] wurde der MILOS-Prototyp in einer nichtverteilten Version demonstriert. Weiterhin ist das System in nationalen und internationalen Fachzeitschriften vorgestellt worden.

Es fand eine systematische Evaluation in Experimenten innerhalb und außerhalb des SFB (u. a. Nortel Networks, Kanada) statt.

3 Durchgeführte Arbeiten und Ergebnisse

Das Teilprojekt bestand aus drei Arbeitsschwerpunkten, wobei Gegenstand des ersten Arbeitsschwerpunkts die Entwicklung eines integrierten Modells zur Prozessunterstützung war. Ein zweiter Arbeitsschwerpunkt bestand aus einer Verfeinerung der im ersten Arbeitsschwerpunkt definierten Konzepte und resultierte in der Implementation des MILOS-Systems. Hierbei waren sowohl ein Modellierungswerkzeug als auch eine Prozessmaschine zu entwickeln, die in einem dynamischen Zusammenspiel die Verzahnung von Projektplanung und -abwicklung ermöglichen. In einem dritten Arbeitsschwerpunkt sollte anhand von Software-Entwicklungsprozessen aus dem Anwendungsgebiet „Gebäudeautomation" der Nutzen des Werkzeugs für technisch- und management-orientierte Rollen experimentell bestimmt werden. Dies erforderte den Einsatz von MILOS im Rahmen von Fallstudien und kontrollierten Experimenten.

3.1 Arbeitsschwerpunkt A: Konzeption eines integrierten Ansatzes

In diesem Arbeitsschwerpunkt wurden zunächst die Sprache MILOS weiterentwickelt [8] [12] [51] und die Prozessmaschine konzipiert (Arbeitspaket AP-A1), was als Grundlage für alle weiteren Arbeiten dienten. Die Konzeption basierte dabei auf den von beiden Arbeitsgruppen gemachten Erfahrung bei der Entwicklung von MVP-L [46] und CoMo-Kit [48]. Im Rahmen der Arbeitspakete AP-A2 und AP-A3 wurden dann allgemeine kausale Abhängigkeiten ermittelt, die für lokale und globale Umplanungsaktivitäten relevant sind [42] [43]. Eine lokale Umplanungen ist z. B. die Verfeinerung eines Prozesses während der Ausführung. Globale Umplanungsaktivitäten finden bei sich ändernden Rahmenbedingungen statt (z. B. Änderungen an der Menge der zur Verfügung stehenden Ressourcen). Die Operationalisierung dieser und zusätzlich vom Benutzer spezifizierbarer Abhängigkeiten bildet die Grundlage des im weiteren Verlauf des Teilprojekts in Zusammenarbeit mit Teilprojekt B2 entworfenen Notifikationsmechanismus.

3.2 Arbeitsschwerpunkt B: Realisierung einer flexiblen Modellierungs- und Ausführungsumgebung für Software-Entwicklungsprozesse

Ziel dieses Arbeitsschwerpunkts war, aufbauend auf den im ersten Arbeitsschwerpunkt entworfenen Konzepten ein Werkzeug zu implementieren, welches Modellierung, Planung und Ausführung von Projekten integriert. Hierzu wurde, ausgehend von den in Arbeitspaket AP-B1 aufgestellten objektorientierten Anforderungsspezifikationen, in Arbeitspaket AP-B2 der Entwurf des Systems vorgenommen [22]. Dabei wurde versucht, die existierenden Defizite des CoMo-Kit Modelers bereits in dieser Phase zu vermeiden. Trotzdem stellte sich im Laufe des Projekts heraus, dass viele technische Aspekte, wie z. B. die Transaktionssemantik des OODBMS GemStone, zum Zeitpunkt der Implementierung (Arbeitspakete AP-B3, AP-B4, AP-B5 und AP-B7) die Anpassung des Systemsentwurfs erforderten.

Im Rahmen des Arbeitspakets AP-B3 entstand in Zusammenarbeit mit Teilprojekt B2 auch eine Anbindung an das kommerzielle Werkzeug Project von Microsoft [20] [38]. Dies erfolgte einerseits, um dem Benutzer von MILOS eine vertraute Arbeitsumgebung zu präsentieren, andererseits um die Tauglichkeit der im System-Entwurf (Arbeitspaket AP-B2) vorgesehenen Schnittstellen zu kommerziellen bzw. im SFB entwickelten Software-Entwicklungswerkzeugen zu demonstrieren. Die dort ebenfalls geforderte Schnittstelle zur Erfahrungsdatenbank des SFB wurde zwar im Entwurf berücksichtigt, konnte aber aus Zeitgründen in der Implementierung noch nicht umgesetzt werden. Dies ist zu Beginn der nächsten Förderungsperiode in Zusammenarbeit mit den Teilprojekten A1, A2 und A3 vorgesehen.

Die Verwaltung von Abhängigkeiten [44] ist eine notwendige Voraussetzung für eine effiziente und korrekte Umplanung. Traditionell wird dies durch bestimmte Ausprägungen von Truth Maintenance Systemen realisiert und lag auch dem CoMo-Kit System in der Vergangenheit zugrunde. Das in Teilprojekt B2 entwickelte und in MILOS einheitlich verwendete Konzept der Event-Condition-Action-Regeln [43] erlaubt eine direkte Abbildung der hier betrachteten Abhängigkeiten als Beziehung zwischen Events und den in den Condition formulierten Constraints. Somit wurde im Rahmen des Arbeitspakets AP-B6 der dort vorgesehene Mechanismus zur Verwaltung von Abhängigkeiten im Zusammenarbeit mit Teilprojekt B2 in MILOS integriert. Dieser Mechanismus bildete die Grundlage für den ebenfalls in diesem Arbeitspaket zu implementierenden Notifikationsmechanismus.

Die notwendigen System- und Integrationstests (Arbeitspaket AP-B8) wurden parallel zur Entwicklung von studentischen Gruppen in Kaiserslautern und Calgary (Kanada) vorgenommen.

3.3 Arbeitsschwerpunkt C: Evaluierung des Prototypen

Ziel dieses Arbeitsschwerpunkts ist die experimentelle Erprobung der Modellierungs- und Ausführungsumgebung MILOS. Zu untersuchende Aspekte sind Fragen nach einer Reduktion des Aufwands für Umplanungen oder einer korrekten Benachrichtigung von Projektmitarbeitern bei Entscheidungsänderungen. Der Nutzen für technisch- und management-orientierte Rollen steht dabei im Vordergrund.

Mittels eines kontrollierten Experiments wurde der MILOS-Prototyp untersucht. Aus Gründen der Laufzeitstabilität des MILOS Systems wurde zunächst auf die nichtverteilte Version zurückgegriffen. Die Untersuchung wurde angelegt, um den integrierten Benachrichtigungsmechanismus und die prozess-gekoppelte Messunterstützung von MILOS zu evaluieren. Der Aufbau des Experiments erfolgte nach der in Teilprojekt B1 erstellten Experimentiermethodik [49], die sich am GQM-Paradigma [45] orientiert.

Das übergeordnete Ziel der Studie entsprechend dem GQM-Schema:

Analysiere die Modellierungs- und Ausführungsumgebung MILOS
zum Zwecke des Evaluierens
bezüglich der Unterstützung bei Umplanungen während der Projekt-Durchführung
aus dem Blickwinkel von Projektmanagern
im Kontext eines kontrollierten Experiments an der Universität Kaiserslautern.

Folgende Fragen ergeben sich aus diesem Ziel, die mittels des Experiments beantwortet werden sollten: (a) Bestimmt der integrierte Notifikationsmechanismus von MILOS die Menge der von einer Umplanung betroffenen Projektmitglieder mit weniger Fehlern als ein Projektplaner ohne technische Unterstützung von MILOS? (b) Wird durch den integrierten Notifikationsmechanismus von MILOS der Aufwand für Umplanungen gegenüber einer vom Projektmanagement organisierten und durchgeführten Benachrichtigung (d. h. ohne MILOS-Unterstützung) stark reduziert? (c) Werden durch die prozess-gekoppelte Unterstützung von MILOS die Messdaten weniger unvollständig und eher entsprechend den Vorgaben im Messplan erhoben als bei deren Erfassung mittels Fragebögen ohne Unterstützung durch MILOS? Zur Überprüfung der Hypothesen und zur Ableitung geeigneter Maße erfolgte eine Verfeinerung der Fragen, wie es im GQM-Ansatz vorgesehen ist.

Bis zum Ende der Förderungsperiode sind weitere Experimente geplant, die auch die verteilte Version von MILOS zum Gegenstand haben werden. So sollen einerseits die Signifikanz der gewonnenen Ergebnisse erhöht und andererseits weitere Aspekte (wie z. B. deskriptive Modellierung) untersucht werden.

4 Zusammenarbeit mit anderen Teilprojekten

Das diesem Teilprojekt zu Grunde liegende Modell der Prozessunterstützung basiert auf der Prozessmodellierungssprache MILOS, die den in Teilprojekt B1 entwickelten MVP-Ansatz mit dem in Teilprojekt B2 entstandenen CoMo-Kit-Ansatz verbindet. Zusammen mit den Teilprojekten A1 und A3 wurde die Anbindung MILOS an die SFB-EB eingeleitet. Die Evaluierung des Werkzeug-Prototypen erfolgte nach der Experimentiermethodik, die in Teilprojekt B1 definiert wurde.

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] J. Bonnermann: *Ressourcenmanagement in MILOS*; Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2000

[2] S. Christofidelli: *Zustandsmodelle in flexiblen Entwicklungsprozessen;* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[3] J. Hohwiller: *Building a virtual marketplace for software development tasks;* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2000

[4] J. Petry: *Vor- und Nachbedingungen für Prozesse in MILOS;* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[5] H. Pharo: *Modellierung von Meta-Aufgaben in MILOS*; Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2000

[6] A. Raquet: *Spezifiaktion und Integration von Vor- und Nachbedingungen für Aufgaben in CoMo-Kit*; Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1997

[7] B. Schadt: *Realisierung eines Frameworks zur graphischen Darstellung und Editierung von MILOS-Objekten in Java*; Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2000

Diplomarbeiten

[8] C. Arenz: *Modellierung von Softwareengineering-Prozessen mit MILOS;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[9] U. Balg: *Wiederverwendung von Entwurfsmustern durch ein Workflow-Management-System;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[10] A. Herzlinger: *A Distributed Workflow Management Approach for Process-oriented Knowledge Management Systems;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[11] S. Mungenas: *Entwicklung eines Ansatzes zur verteilten Abarbeitung von Workflows in einem virtuellen Unternehmen*; Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[12] F. Neblung: *Validierung und Erweiterung der Prozeßmodellierungssprache MILOS;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[13] A. Raquet: *Dynamic Project Management for Distributed Construction Processes;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[14] T. Rüger: *Behandlung von Risiken in CoMo-Kit;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[15] M. Schaaf: *Ein System zum Kommunikationsmanagement verteilter Objekte in Java;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[16] M. Schlindwein: *A Framework Using Computer Agents to Provide Automatic Execution Support for Processes in Software Development Projects;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[17] S. Schmitt: *WWW-Based Software Process Modeling with CoMo-Kit*; Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

Vorträge

[18] B. Dellen: *Internetbasiertes Wissensmanagement in Softwareprojekten*; Wissensmanagement 1999: Strategien der Gegenwart, Industriekonferenz, Frankfurt, 1998

[19] S. Goldmann: *Internetbasierte Prozessmodellierung und Projektplanung*; Wissensmanagement 1999: Strategien der Gegenwart, Industriekonferenz, Frankfurt, 1998

Tagungsbeiträge

[20] F. Bendeck, S. Goldmann, H. Holz, B. Kötting: *Coordinating Management Activities in Distributed Software Development Projects*; Proceedings of the 7th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 33-38, 1998

[21] B. Dellen, F. Maurer: *Internet Based Software Process Support*;
Proceedings of the 7th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 27-32, 1998

[22] B. Dellen, F. Maurer: *A Concept for an Internet-based Process-Oriented Knowledge Management Environment*; Proceedings of the KAW'98, Banff, Canada, 1998

[23] S. Goldmann, J. Münch, H. Holz: *Distributed Process Planning Support with MILOS*; Proceedings of the 11th International Software & Engineering and Knowledge Engineering Conference (SEKE ‘99), Knowledge Systems Institute, ISBN 1-891706-01-2, 1999

[24] S. Goldmann, B. Kötting: *Working Group Report of the 2nd Workshop on Coordinating Distributed Software Development Projects*; Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '99), IEEE Computer Society Press, 1999

[25] S. Goldmann, J. Münch, H. Holz: *A Meta-Model for Distributed Software Development*; Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '99), IEEE Computer Society Press, 1999

[26] S. Goldmann, , J. Münch, H. Holz: MILOS: *A Model of Interleaved Planning, Scheduling, and Enactment;* presented at the ICSE '99 Workshop on „Software Engineering over the Internet"; Web proceedings only: http://sern.cpsc.ucalgary.ca/~maurer/ICSE99WS/Program.htm

[27] H. Holz, S. Goldmann, F. Maurer: *Working Group Report on Coordinating Distributed Software Development Projects;* Proceedings of the 7th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 69-72, 1998

[28] B. Kötting, F. Maurer: *Approaching Process Support for Virtual Software Corporations;* presented at the ICSE '99 Workshop on Software Engineering over the Internet; Web proceedings only: http://sern.cpsc.ucalgary.ca/~maurer/ICSE99WS/Program.htm

[29] B. Kötting, F. Maurer: *A Concept for Supporting the Formation of Virtual Corporations through Negotiation;* Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '99), IEEE Computer Society Press, 1999

[30] F. Maurer, H. Holz: *Process-Oriented Knowledge Management For Learning Software Organizations;* Proceedings of the 12th Knowledge Acquisition Workshop (KAW '99), Banff, Canada, 1999

[31] F. Maurer, G. Succi, H. Holz, B. Kötting, S. Goldmann, B. Dellen: *Software Process Support over the Internet;* Proceedings of the 21th International Conference on Software Engineering (ICSE '99), ACM Press, ISBN 1-58113-074-0, pp. 642-645, 1999

[32] F. Maurer, H. Holz: *Process-Centered Knowledge Organization for Software Engineering;* Aha, D.W.,& Munoz Avila, H. (Eds.): Exploring Synergies of Knowledge Management and Case-Based Reasoning: A 1999 AAAI Workshop (Working Notes). (Technical Report AIC-99-008). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, 1999

[33] 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

[34] Ch. Petrie, S. Goldmann, A. Raquet: *Agent-Based Project Management;* Proc. of the Intnl.Workshop on Intelligent Agents in CSCW, Deutsche Telekom, Dortmund, Sept. 1998, p. 1-1. (short version of paper of the same name in Springer LNAI 1600)

[35] M.M. Richter, B. Kötting, S. Goldmann: *Flexibles Workflow Management in Software Development Processes;* to appear in Proceedings of the "Forschungsforum Wissensmanagement und schnelle Produktentwicklung", Stuttgart, Germany, 1999

Zeitschriften / Monografien

[36] S. Goldmann, B. Kötting: *Software Engineering over the Internet;* IEEE Internet Computing, Vol. 3, No 4, July/August 1999, pp 93 – 94, IEEE Computer Society Press, 1999

[37] S. Goldmann, J. Münch, H. Holz: *Distributed Process Planning Support with MILOS;* Int. Journal of Software Engineering and Knowledge Engineering, October 2000 (to appear)

[38] F. Maurer, B. Dellen, F. Bendeck, S. Goldmann, H. Holz, B. Kötting, M. Schaaf: *Merging project planning and web-enabled dynamic workflow for software development;* IEEE Internet Computing, Special Issue on Internet-Based Workflow, May/June 2000 (to appear)

[39] Ch. Petrie, S. Goldmann, A. Raquet: *Agent-Based Project Management;* Springer Lecture Notes on Artificial Intelligence (LNAI) 1600 special volume on "Artificial Intelligence Today", Springer Verlag, 1998

[40] M. M. Richter: *Knowledge-Based Systems;* Intelligence and Artificial Intelligence (ed. U. Ratsch, M.M. Richter, I.-O. Stamatescu), Springer Verlag, 1998

6 Fremdveröffentlichungen

[41] Frank Maurer. *Hypermedia-based Knowledge Engineering for distributed, knowledge-based Systems;* Dissertation, Universität Kaiserslautern, 1993

[42] B. Dellen, F. Maurer: *Change Impact Analysis Support for Software Development Processes*; Journal of Applied Software Technology, International Academic Publishing, 1998

[43] Dellen, Barbara: *Change Impact Analysis Support for Software Development Processes*; Dissertation, Universität Kaiserslautern, Kaiserslautern, Feb. 2000

[44] Barbara Dellen, Frank Maurer, Jürgen Paulokat: *Verwaltung von Abhängigkeiten in kooperativen wissensbasierten Arbeitsabläufen*; Richter/Maurer (Hrsg.): Expertensysteme 95, infix Verlag, 1995

[45] V. R. Basili, G. Caldiera, H. D. Rombach: *Goal Question Metric Paradigm*; In John J. Marciniak, editor, Encyclopedia of Software Engineering, Vol. 1, pp. 528-532, John Wiley & Sons, 1994

[46] Alfred Bröckers, Christopher M. Lott, H. Dieter Rombach, Martin Verlage: *MVP-L Language Report Version 2*; Technischer Bericht Nr. 265/95, Universität Kaiserslautern, 1995

[47] Barbara Dellen, Frank Maurer, Jürgen Paulokat: *Verwaltung von Abhängigkeiten in kooperativen wissensbasierten Arbeitsabläufen*; Richter/Maurer (Hrsg.): Expertensysteme 95, infix Verlag

[48] Sigrid Goldmann: *CoMo-Kit und MVP-L: Ein Vergleich zwischen zwei Werkzeugen zur Prozeßmodellierung und -ausführung*; Interner Bericht Nr. 02/1995, Sonderforschungsbereich 501, Universität Kaiserslautern, 1995

[49] Oliver Laitenberger, Jürgen Münch: *Ein Prozeßmodell zur experimentellen Erprobung von Software-Entwicklungsprozessen;* SFB 501, Technischer Bericht 04/1996, Fachbereicht Informatik, Universität Kaiserslautern, 1996

[50] Martin Verlage, Barbara Dellen, Frank Maurer, Jürgen Münch: *A Synthesis of Two Process Support Approaches*; Proceedings of the Eighth International Conference on Software Engineering and Knowledge Engineering (SEKE`96), Lake Tahoe, Nevada, USA, 1996

[51] Boris Kötting. *Konzeption und Implementierung einer Sprache für Software-Entwicklungsprozesse*; Diplomarbeit, Universität Kaiserslautern, Fachbereich Informatik, 1996

[52] Barbara Dellen, Frank Maurer, Jürgen Münch, Martin Verlage: *Enriching Software Process Support by Knowledge-based Techniques*; International Journal of Software Engineering and Knowledge Engineering, 1997

Teilprojekt A3

Unterstützung des Softwareentwicklungsprozesses durch objekt-relationale Datenbanktechnologie

Leiter

Prof. Dr. Theo Härder
geb. 28.08.45
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-4030
Fax: 0631/205-3299
haerder@informatik.uni-kl.de

Dr. Norbert Ritter
geb. 07.10.62
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-3264
Fax: 0631/205-3299
ritter@informatik.uni-kl.de

Mitarbeiter

Ergänzungsausstattung:

Dipl. Inform. U. Marder
Dipl. Inform. W. Mahnke

Grundausstattung:

Dipl. Inform. H.-P. Steiert
M. Sc. Nan Zhang
Dipl.-Inform. Weiping Zhang

1 Kurzfassung der Aufgabenstellung

Die Integration objektorientierter und relationaler Konzepte im Rahmen von objekt-relationalen Datenbankverwaltungssystemen (ORDBVS) [53] verspricht bei erster Betrachtung, dass viele der (Datenverwaltungs-) Anforderungen von SE-Prozessen durch Einsatz dieser neuen Generation von Datenbanktechnologie erfüllt werden können. So erlauben die erweiterten Modellierungsmöglichkeiten sowie vor allem die Erweiterbarkeitseigenschaft, dass sowohl einfach als auch komplex strukturierte Daten sowie Daten unterschiedlichster Formate in integrierter Weise verwaltet werden können. Desweiteren können extern gespeicherte Daten in die DB-Verarbeitung miteinbezogen werden; so lässt sich in einfacher Weise ein Web-Zugriff ermöglichen. Die Nutzbarmachung und Evaluierung dieser neuen Technologie im Rahmen von Softwareentwicklungsumgebungen war und ist eines der Ziele dieses Teilprojekts.

Angesichts der Tatsache, dass ORDBVS, obwohl mittlerweile eine Standardisierung der Schnittstelle mit dem SQL:99-Standard [54, 55] erreicht worden ist, immer noch in der Entwicklung sind, soll dieses Teilprojekt auch diese Entwicklung mit beeinflussen. Dies bedeutet im wesentlichen, dass zunächst Prinzipien/Methoden der Nutzung von Erweiterbarkeit bei der Anwendungsentwicklung und Architekturansätze für ORDBVS-basierte Anwendungssysteme entwickelt werden müssen. Weiterhin ist das rein Server-zentrierte Verarbeitungsmodell, das ORDBVS von den RDBVS geerbt haben, in flexibler Weise um eine Client-basierte Verarbeitung [51] zu erweitern, da letztere in Entwurfsanwendungen benötigt wird.

Diese Ziele wurden im Rahmen von drei Arbeitsschwerpunkten bearbeitet:

- *Grundlagen objekt-relationaler Datenbanktechnologie*
  Es sind Erweiterungsmechanismen untersucht und evaluiert sowie ein Prototyp eines Benchmarks für ORDBVS und Basiskonzepte für die Handhabung multimedialer Daten in ORDBVS entwickelt worden.
- *Anwendungsentwicklung und ORDBVS-basierte Anwendungssysteme*
  Hier sind grundlegende Vorgehensweisen für die generische Entwicklung von ORDB-Anwendungen sowie ein flexibles Verarbeitungsmodell für ORDBVS erarbeitet worden.
- *Direkte Unterstützung des SE-Prozesses und Datenkontrolle*
  Dieser Arbeitsschwerpunkt beschäftigte sich im wesentlichen mit der Entwicklung eines Prototypen einer ORDBVS-gestützten Erfahrungsdatenbank, die mit dem Beginn der kommenden Förderungsperiode zur SFB-weiten Nutzung zur Verfügung gestellt werden soll.

2 Entwicklung des Teilprojekts in der Förderperiode

Die Förderungsperiode 1998-2000 ist die erste Förderungsperiode diese Teilprojekts. Im Laufe dieser Periode stellte sich heraus, dass zwei wesentliche Forschungsrichtungen zu bearbeiten sind, die auch die (weitere) Stellung im Gesamt-SFB betreffen. Die erste dieser Richtungen hat eher Infrastrukturcharakter und ist maßgeblich für die Einordnung des Teilprojektes in den Projektbereich A verantwortlich. Hier geht es um die effektive Unterstützung von SFB-Experimenten durch objekt-relationale Datenbanktechnologie. In diesem Sinne wurde ein Prototyp einer Erfahrungsdatenbank (in Zusammenarbeit mit dem Teilprojekt A1) entwickelt, der in der kommenden Periode durch Integration von Produktdatenverwaltung und Ablaufsteuerung zu einem umfassen SE-Repository ausgebaut werden soll. Die zweite dieser Richtungen beschäftigt sich mit der Nutzung von Erweiterbarkeit zur Entwicklung von ORDBVS-gestützten Anwendungssystemen. Hier geht es um die Erarbeitung geeigneter (generischer) Methoden zur Anwendungsentwicklung und geeigneter Architekturansätze für solche Anwendungssysteme. Darüber hinaus werden neue Konzepte für ORDBVS, wie z. B. ein neues flexibles Verarbeitungskonzept entwickelt, die zur Erfüllung entsprechender Anforderungen der Softwareentwicklung gebraucht werden.

3 Durchgeführte Arbeiten und Ergebnisse

3.1 Arbeitsschwerpunkt A: Grundlagen objekt-relationaler Datenbanktechnologie

Dieser Arbeitsschwerpunkt beschäftigte sich mit eher grundlegenden Aspekten objekt-relationaler Datenbanktechnologie, sodass die bisher erzielten Ergebnisse eher mittelbar in die Unterstützung von SE-Prozessen einfließen. Als grundlegende Aspekte wurden Erweiterbarkeit und Leistungsverhalten von ORDBVS betrachtet.

Die Möglichkeiten der Erweiterbarkeit wurden im ORIENT-Projekt untersucht. Hierzu wurden semantische Beziehungstypen dahingehend als Beispiel betrachtet, dass versucht wurde, Mechanismen zur Kontrolle der Semantik mittels Erweiterbarkeit in das DBVS zu integrieren. Das VirtualMedia-Projekt hingegen betrachtet multimediale Datentypen als Gegenstand der Erweiterbarkeit. Hinsichtlich der Beurteilung des Leistungsverhaltens von ORDBVS wurde ein Benchmark-Prototyp entwickelt. Weiterhin wurde untersucht, wie Referenzen bei der Verarbeitung deskriptiver Anfragen (DBVS-intern) effizient verarbeitet werden können.

ORIENT

Datenbeziehungen spielen bei der präzisen und problemadäquaten Modellierung sowie bei der Verarbeitung komplexer Objekte, wie sie in modernen Anwendungen (z. B. Softwareentwicklung) häufig vorkommen, eine wichtige Rolle. Die bisherigen Datenmodelle und -systeme unterstützen diese Beziehungen nur unzureichend. Daher ist es Ziel von *ORIENT* (Object-based Relationship Integration ENvironmenT), geeignete Mechanismen zur Verfügung zu stellen, mit deren Hilfe Beziehungssemantik durchgängig von der Modellierung bis zur späteren Anwendungsverarbeitung unterstützt und automatisch vom System garantiert werden kann.

Bisher wurden, auf Basis der Untersuchung von verschiedenen Beziehungstypen in Anwendungsszenarien und der Erfassung der darin enthaltenen Semantik, grundlegende Konzepte zur Erweiterung von ORDBS um semantisch reichhaltigere Beziehungstypen entwickelt. Dazu gehören u. a. Eigenschaften zur Ausgestaltung von Beziehungstypen sowie neue bzw. redefinierte Operationen zur systemkontrollierten Manipulation von Beziehungsinstanzen. Dies hat zu der DB-Sprache *OrientSQL* geführt [42, 9], die den gegenwärtigen DB-Standard SQL:99 entsprechend erweitert.

Im Mittelpunkt der Arbeit steht die Integration der Beziehungen, ihrer Operationen sowie der Kontrolle ihrer Semantik. Eine geeignete Basis dafür sollen ORDBVS bieten, die eine hohe Ausdrucksmächtigkeit und Erweiterbarkeit aufweisen. In diesem Zusammenhang wurden drei verschiedene Ansätze im Detail untersucht [43, 9].

- Bei der flachen Integration über eine Zusatzebene (ohne Änderung der herkömmlichen SQL-Schnittstelle) handelt es sich eher um eine rein syntaktische Umsetzung, bei der das Ziel einer semantischen Anreicherung von Beziehungen verfehlt wird.
- Die Tiefenintegration in ein ORDBS dagegen garantiert eine nahtlose Umsetzung der neuen Konzepte und ihre effiziente Verarbeitung. Jedoch impliziert dieser Ansatz weitreichende Veränderungen und Anpassungen der internen Verarbeitungsmechanismen eines DBS, was selbst mit Hilfe der vielversprechenden OR-Erweiterbarkeit bisher sehr schwer zu erreichen ist.
- Die Kompromisslösung für die Integration sieht deshalb die Nutzung von vorgegebenen Systemschnittstellen vor, die in heutigen ORDBS als DataBlades o. ä. angeboten werden. Bei diesem Ansatz kann man nicht von einer Tiefenintegration sprechen, da für eine spezielle Erweiterungsmaßnahme keine zugeschnittenen Systemanpassungen, was interne Funktionen oder Datenstrukturen angeht, eingebracht werden. Die anwendungsbezogene Erweiterung erfolgt eher als „nahe Kopplung“, wodurch aber ein zufriedenstellendes Leistungsverhalten zu erwarten ist.

Durch diese Untersuchungen wurde auch verdeutlicht, welche Eigenschaften eine Erweiterbarkeitsinfrastruktur (insbesondere von ORDBS) aufweisen muss, um *ORIENT* und auch andere von Anwendungen erwünschte Konzepte geeignet und effizient zu unterstützen und zu integrieren [9, 43].

VirtualMedia

Im VirtualMedia-Projekt [16] wird eine Integration von ORDBVS und Medienservern angestrebt, welche die Verwaltung großer, (aus DBVS-Sicht) unstrukturierter Dokumente durch das DBVS sehr viel flexibler und mächtiger macht, als dies bei heutigen DBVS der Fall ist. Im Einzelnen sollen folgende Eigenschaften sichergestellt werden: physische Datenunabhängigkeit (Formatunabhängigkeit), die Möglichkeit, Operationen auf den Dokumenten im DBVS ausführen zu lassen, Verhinderung von Informationsverlusten (verursacht durch konkurrierende Anwendungen), dynamische Optimierung (z. B. durch Transformation von Operatorgraphen, redundante Materialisierung oder Parallelisierung/Pipelining von Operationen) und Echtzeitunterstützung (für multimediale Dokumente). Zusammengefasst werden diese Eigenschaften als „Transformationsunabhängigkeit“ bezeichnet, da sie es Anwendungen ermöglichen, von anderen Anwendungen generierte Dokumente zu öffnen und zu bearbeiten, ohne deren Speicherungsformat zu kennen. Hiervon können vor allem Datenbanken profitieren, bei denen die verwalteten Dokumente in aller Regel wesentlich langlebiger sind als die Anwendungen (z. B. Digitale Bibliotheken). Um mit diesem Wandel mithalten zu können, ist VirtualMedia als ein Framework konzipiert, in dem nur wenige zentrale Algorithmen und Schnittstellen fest vorgegeben sind. Anwendungs(domänen)spezifische Operationen können jederzeit hinzugefügt werden (insbesondere auch nach der Erzeugung der Dokumente, für die sie gedacht sind). Dementsprechend flexibel und in weiten Grenzen erweiterbar ist auch die Anwendungsschnittstelle (in Form einer deskriptiven Anfragesprache) ausgelegt.

Die bisherigen Arbeiten fokussierten vor allem auf die Schaffung eines theoretischen Modells der Server-seitigen Verarbeitung multimedialer Dokumente und die Ableitung einer für einen solchen Server geeigneten abstrakten Anwendungsschnittstelle. Auf dieser Grundlage wurden weiterhin Konzepte für eine Semantik-basierte Optimierung von Zugriffs- und Verarbeitungsprozessen auf der Serverseite erarbeitet. Überlegungen zu möglichen Architekturen zur Realisierung eines MM-DBVS mit VirtualMedia (Referenzarchitektur, Implementierungsarchitekturen) wurden ebenfalls angestellt. Die Mehrzahl der genannten Aspekte ist in [15] bzw. [33] ausführlicher dargestellt. Es wurde ferner damit begonnen, Einsatzmöglichkeiten des VirtualMedia-Konzepts im Kontext der Erfahrungsdatenbank (EDB, s. Abschnitt 3.3) auszuloten. In der EDB sollen Dokumente, die von externen SE-Werkzeugen erzeugt und bearbeitet werden, in ihrem proprietären Format verwaltet werden. Um diese Daten auch mit einfacheren Werkzeugen, z. B. Browsern, auswerten zu können, ist derzeit vorgesehen, verschiedene physische Repräsentationen dieser Daten zu speichern. Die durch VirtualMedia realisierte Formatunabhängigkeit kann dieses Problem freilich wesentlich eleganter lösen, indem die Dokumente je nach Bedarf der anfragenden Anwendung in der jeweils optimalen externen Repräsentation ausgeliefert werden, ohne alle möglichen Repräsentationen physisch vorzuhalten zu müssen.

OR-Benchmark

Eine Softwareentwicklungsumgebung ist ein einfaches Beispiel dafür, daß DBVS heutzutage üblicherweise als Komponenten großer Softwaresysteme auftreten. Sie dienen innerhalb dieses Verbunds von Komponenten meistens als persistente Objektsysteme. Da, aufgrund der bisher unzureichenden Integration objektorientierter Konzepte in das relationale Modell, gegenwärtige ORDBVS nicht die von den übergeordneten Komponenten geforderten, objektorientierten Konzepte anbieten, wird eine Zwischenschicht benötigt, die diese Lücke schließt.

Zur Beurteilung des Leistungsverhaltens eines solchen Systems ist es nun notwendig, nicht nur das Leistungsverhalten des DBVS, sondern auch den durch die Abbildungsschicht verursachten Aufwand zu betrachten. Insgesamt müssen die folgenden beiden Fragen beantwortet werden: Welcher (Mindest-)Aufwand wird von der Abbildungsschicht hinsichtlich der Abbildung der Objektstrukturen (Schematransformation) und der entsprechend notwendigen Transformation von Anfragen verursacht? Welches

Leistungsverhalten zeigt das genutzte DBVS in der Verarbeitung typischer, von der Abbildungsschicht (dynamisch) erzeugter SQL-Anweisungen?

Während die zweite Frage (vom Ansatz her) mit herkömmlichen Benchmarksystemen beantwortet werden kann, ist zur Einbeziehung der ersten Frage ein Überdenken der traditionellen Benchmark-Ansätze unerlässlich. In [44, 45] wird ein Prototyp eines Benchmarksystems vorgestellt, das die Beurteilung des Leistungsverhaltens von ORDBVS in dem oben angesprochenen Anwendungsszenario erlaubt.

Parallele Navigation

Referenzen und mehrwertige Attribute sind neben der Einführung von benutzerdefinierten Typen und Typ-/Tabellenhierarchien die Schlüsselkonzepte, die für die „Objektorientierung“ objekt-relationaler Datenmodelle verantwortlich zeichnen. Im Vergleich zum Relationenmodell, bei dem alle Objekte und Beziehungen ausschließlich durch Werte dargestellt sind und deshalb auch Suche, Zugriff und Verknüpfung wertbasiert erfolgen, ergibt sich insbesondere durch den Einsatz von einfachen und mengenwertigen Referenzen ein wesentlich größeres Modellierungs- und Realisierungsspektrum für die Darstellung und den Zugriff. Exemplarisch lässt sich das anhand von (n:m)-Beziehungen belegen, die im Gegensatz zum Relationenmodell direkt und symmetrisch (etwa durch SETOF (REF ...)) abgebildet werden können.

Eine Kernfrage unserer Arbeiten lautete deshalb, wie diese größere Darstellungsflexibilität bei der Auswertung von Anfragen (Selektion, Verbund) genutzt werden kann. Im Kontext von komplexen Objekten haben wir diese Frage systematisch untersucht. Die Auswertung von Referenzen (Dereferenzierung) gestattet systemseitig den Einsatz von Navigationsalgorithmen bei der Ableitung von Anfrageergebnissen. Bei der Durchführung der Selektion erlaubt der direkte Zugriff auf einzelne Objekte verbesserte Möglichkeiten der Prädikatsauswertung (EXISTS, FOR ALL). Außerdem bieten sich bei der Ausführung der zugrundeliegenden Operatorbäume größere Spielräume zur Optimierung an, da sich neuartige Auswertungsstrategien, die wir „top-down evaluation“ und „bottom-up restriction“ genannt haben, anwenden lassen. Auch für die Verknüpfung von Objektmengen ergeben sich neben dem wertbasierten Verbund neue auf REF-Typen beruhende Auswertungsmöglichkeiten. Wir entwickelten dazu einen sog. CONNECT-Operator, der neben dem direkten Objektzugriff (bei REF) auch verschiedene Algorithmen zur parallelen Navigation (bei SETOF (REF)) ausnutzen kann. Die Ergebnisse unserer Arbeiten zu diesem Themenbereich sind vor allem in [28] zusammengefasst.

3.2 Arbeitsschwerpunkt B: Anwendungsentwicklung/ORDBS-basierte Anwendungssysteme

In diesem Arbeitsschwerpunkt wurden Methoden der Entwicklung von ORDB-Anwendungen und zugehörige Architekturansätze entwickelt. Das SERUM-Projekt stellt zur Unterstützung dieses Entwicklungsprozesses ein Framework bereit, dass den Entwickler mittels generischer Mechanismen weitgehend entlastet. Ebenfalls auf generischen Mechanismen beruht das Prinzip der generierten Aufrufschnittstellen, das dem Anwendungsprogrammierer auf die benötigten DB-Zugriffe zugeschnittene Funktionen zur Manipulation von DB-Daten in objektorientierten Programmiersprachen bietet. Neben den angesprochenen Methoden der Anwendungsentwicklung wurde in diesem Arbeitsschwerpunkt auch ein neues Verarbeitungsmodell entwickelt, das die bisherige Server-zentrierte Verarbeitung von ORDBVS um eine Client-basierte Verarbeitung erweitern soll.

SERUM

Gegenüber den herkömmlichen RDBVS bieten ORDBVS eine stark erweiterte Funktionalität. Bisher besteht noch wenig Erfahrung darüber, wie diese in den verschiedenen Anwendungsdomänen verwendet werden muss, damit DB-gestützte Applikationen bezüglich Entwicklungszeiten, Leistungsverhalten und Funktionalität davon profitieren.

Ziel unseres SERUM-Projektes ist es, Richtlinien für die Realisierung DB-gestützter Anwendungen zu entwickeln. Aufgrund der gestiegenen Vielfalt von Möglichkeiten beim Einsatz eines ORDBVS ist die Anwendungsentwicklung deutlich komplexer geworden. Daher soll das im Projekt zur Verfügung stehende Spezialistenwissen genutzt werden, um den Entwicklungsprozess weitgehend zu automatisieren. Dazu wurde zunächst ein Vorgehensmodell [31, 29] erarbeitet, das den SERUM-Prozess grob in die vier Phasen Spezifikation, Verfeinerung, Konfiguration und Generierung untergliedert. In der ersten Phase werden anwendungsspezifische Aspekte der zu entwickelnden Datenhaltungskomponente mittels der objektorientierten Modellierungssprache UML. Grundlage der Verfeinerung sind nun vorgefertigte Modellierungsmuster, mit denen werkzeugunterstützt das initiale Modell erweitert wird. Für die vorgefertigten Konstrukte existieren bereits halbfertige Realisierungen, die durch Parametrisierung und Codegenerierung vervollständigt und mit den anwendungsspezifischen Teilen verbunden werden. Beides zusammen bezeichnen wir als SERUM-Framework. Bei der Realisierung der Spezifikation stehen noch Wahlmöglichkeiten für die verwendete Technologie und die konkrete Abbildung offen. Diese müssen in einer Konfigurationsphase festgelegt werden. Aus der nun vollständigen Spezifikation erzeugt ein Generator letztlich die Datenhaltungskomponente.

Um den Prozess geeignet zu unterstützen, haben wir eine Infrastruktur von Werkzeugen entwickelt. In deren Mittelpunkt steht das UML-Repository in dem die Spezifikation der Datenhaltungskomponente als UML-Modell abgelegt wird. Es dient uns zum einen als notwendiges Werkzeug im SERUM-Prozess, zum anderen als Beispielanwendung. So konnten wir im Rahmen der Entwicklung des Repositories einen grundlegenden Überblick über die Modellierungssprache UML [36, 38] und die Funktionalität von ORDBVS [32] gewinnen. Ein Ergebnis dieser Arbeit sind Entwurfsregeln für die Realisierung von Anwendungen mit ORDBVS, insbesondere für die Abbildung eines UML-Modelles auf ein objekt-relationales Datenbankschema. Diese können nun in einem Schemagenerator implementiert werden. Am Beispiel des Repositories konnten wir weiterhin analysieren, inwieweit die Erweiterungsmöglichkeiten von ORDBVS helfen, fehlende Funktionalität nachzubilden [32]. Beispielhaft sollen hier Objektidentifikatoren und Mehrfachvererbung genannt werden. Weiterhin haben wir die Integration externer Werkzeuge untersucht. Am Beispiel des UML-Repository wurde ein Konzept entwickelt, um mit Mitteln des ORDBVS auf externe Daten, hier die eines graphischen Modellierungswerkzeugs, zuzugreifen. Während der Entwicklung des Repositories konnten wir außerdem konzeptionelle Mängel kommerzieller ORDBVS [32] aufdecken, die zu einem Anforderungskatalog an zukünftige ORDBVS-Generationen führen sollen.

Neben den angesprochenen Aspekten beschäftigten wir uns in SERUM mit der Bereitstellung von Komponenten, die höherwertige Dienste für Entwurfsanwendungen realisieren, wie beispielsweise die Versionierung von Produktdaten [31] oder Unterstützung für Ablaufsteuerung, sowie mit geeigneten Mehr-Ebenen-Architekturen zur Unterstützung von Entwurfsanwendungen [40].

Flexibles Verarbeitungsmodell

ORDBVS erlauben es, Anwendungsfunktionalität nicht nur im Anwendungsprogramm, sondern auch im DBVS auszuführen. Der Aufruf von Anwendungsfunktionen kann dabei auch im Rahmen der Auswertung einer SQL-Anfrage erfolgen. Damit ergeben sich bei der Konzeption der Client/Server-Architektur von Entwurfsanwendungen neue Möglichkeiten. Unser Ziel ist es, die bisher für Entwurfsanwendungen typische 2-Ebenen-Architektur mit hauptsächlich Client-basierter Verarbeitung und statischer Aufteilung von Daten- und Rechenlast aufzuweichen. Statt dessen streben wir eine Mehr-Ebenen-Architektur mit dynamischer Wahl des Ausführungsortes an.

Die Eignung eines Ausführungsortes hängt in datenintensiven Anwendungen wesentlich davon ab, wie aufwendig die benötigten Daten zu diesem Ort zu transportieren sind. Ein Entscheidungsalgorithmus für die Wahl des Ausführungsortes muss aber neben der aktuellen Datenverteilung und den daraus resultierenden Kosten für die Funktionsausführung an einem bestimmten Ort auch die zukünftige Verarbeitung berücksichtigen, da sich daraus ergibt, ob die Daten dort eventuell in naher Zukunft noch einmal benötigt werden. Des weiteren soll sich der Entscheidungsalgorithmus bei Veränderungen der Ablaufumgebung dynamisch an die neue Situation anpassen.

Wir haben uns für einen „black-box"-Ansatz entschieden, der ohne a-priori-Analyse des Anwendungscodes und dynamische Analyse der aktuellen Datenverteilung auskommt. Unser Konzept [1, 5] für einen Entscheidungsalgorithmus stützt sich auf stochastische Methoden. Das Prognosemodell repräsentiert ein stochastisches Modell der bisherigen Verarbeitung, das aus protokollierten Sequenzen von Aufrufen der Anwendungsfunktionen gewonnen wird. Es erlaubt uns, zu einem aktuellen Systemzustand mögliche zukünftige Aufrufsequenzen vorherzusagen und mit Wahrscheinlichkeiten zu belegen. Diese Aufrufsequenzen werden dann für die verschiedenen alternativen Ausführungsorte der Funktionen bewertet. Dazu haben wir ein Kostenmodell entwickelt, das im wesentlichen auf der zur Laufzeit gemessenen Ausführungszeit basiert und periodisch aktualisiert wird. Dieses Konzept haben wir für das Szenario einer 2-Ebenen-Architektur mit einem ORDBMS konkretisiert und dazu eine passende Evaluierungsumgebung konzipiert.

Generierte Aufrufschnittstellen

Zum DB-Zugriff wird eine Anbindung der DB-Sprache (SQL) an die entsprechende (objektorientierte) Programmiersprache benötigt, die auch als Anwendungsprogrammierschnittstelle (kurz *API* für engl. *application programming interface*) bezeichnet wird. Verschiedenartige Formen der Anbindung weisen unterschiedliche Charakteristika auf. So zeichnen sich sogenannte Spracheinbettungen, wie eSQL oder SQLJ, durch höhere Effizienz aber niedrigere Datenunabhängigkeit als sogenannte, traditionelle Aufrufschnittstellen, wie ODBC/JDBC, aus. Mit dem in [37, 48] beschriebenen Ansatz der generierten Aufrufschnittstellen haben wir folgende Ziele erreicht:

- Bereitstellung einer API, die Client-basierte Verarbeitung unterstützt;
- Vereinigung der Vorteile von Einbettungen und Aufrufschnittstellen;
- gezielte Optimierung der Datenzugriffe ausgehend von spezifischen Anwendungsanforderungen.

Diese Ziele konnten insbesondere dadurch erreicht werden, dass für einzelne Checkout-Anweisungen (Retrieval) unterschiedliche Bindungsgrade vom Anwendungsprogrammierer angegeben werden können, die wiederum festlegen, ob die Checkout-Operation selbst und die zur Verarbeitung der Treffermenge (im Client-Puffer) benötigten Operationen früh oder spät gebunden werden. Neben der Wahl des Bindungszeitpunktes kann durch geeignete Konfigurierung von Pointer-Swizzling- und Prefetching-Mechanismen eine weitere Optimierung erreicht werden.

3.3 Arbeitsschwerpunkt C: Datenkontrolle/Basisdienste für die Softwareentwicklung

Dieser dritte Arbeitsschwerpunkt beschäftigte sich vorrangig mit der Nutzung von ORDBVS zur Entwicklung einer sogenannten Erfahrungsdatenbank. Dazu wurde in Zusammenarbeit mit dem Teilprojekt A1 ein Prototyp einer Erfahrungsdatenbank (SFB-501-EDB) entwickelt, der für eine Nutzung im Rahmen von SFB-Experimenten der kommenden Förderungsperiode vorgesehen ist. Da eine integrierte Erfahrungs- und Produktdatenverwaltung angestrebt wurde, mussten hier auch spezifische Mechanismen der Datenkontrolle (Zugriffskontrolle, Kooperationskontrolle) sowie der Versionierung angedacht werden.

Erfahrungsdatenbank

Ausgehend von einem im Teilprojekt A1 entwickelten Ablageschema für Erfahrungsdaten und einem zugehörigen (HTML-)Datei-basierten Prototypen [50] einer Erfahrungsdatenbank wurde in Kooperation mit A1 ein ORDBVS-basierter Prototyp einer Erfahrungsdatenbank entwickelt, der mit dem Beginn der kommenden Förderungsperiode zur SFB-weiten Nutzung freigegeben wird. Erste Versionen dieses Systems wurden bereits auf Tagungen demonstriert.

Zu Beginn (der Entwicklung des ORDBVS-basierten EDB-Prototyps) wurden zunächst die Anforderungen analysiert. Folgende Gründe führten zur Auswahl eines ORDBVS als Implementierungsgrundlage:

- Erweiterte Modellierungsmöglichkeiten werden zur Erfassung von komplex strukturierten Erfahrungs- und Produktdaten benötigt.
- Es müssen Datentypen für große Objekte (Large Objects, LOBs) eingesetzt werden, um verschiedene Repräsentationen von Erfahrungen zu verwalten.
- Extern zu speichernde Daten (Dateien) müssen in die DB-Verarbeitung miteinbezogen bzw. unter Kontrolle des DBVS gestellt werden können.
- Es muss in einfacher Weise eine Web-basierte Schnittstelle implementiert werden können.

Das anschließend entwickelte DB-Schema reflektiert das durch A1 vorgegebene Ablageschema [10], das im wesentlichen eine Klassifikation von Erfahrungen vorgibt. Erfahrungen können in mehreren, verschiedenen Repräsentationen abgelegt werden, zwischen denen jedoch von einer logischen Äquivalenzbeziehung auszugehen ist. Eine Speicherung derartiger Repräsentationen legt die Nutzung von LOB-Datentypen im DBS nahe. Um dennoch eine angemessene ähnlichkeitsbasierte Suche nach Erfahrungen anbieten zu können, wird jede Erfahrung mit einem sogenannten *Charakterisierungsvektor* assoziiert, dessen Inhalte das Erfahrungselement beschreiben und dessen Struktur (Attribute, Beziehungen) so zur Formulierung von Suchkriterien herangezogen werden kann. Zur Verwaltung von Produktdaten werden ähnliche Strukturen bereitgestellt. Dabei ist jedoch zu beachten, dass die laufenden Projekten zugeordneten Produktdaten einer eingeschränkten Sichtbarkeit unterliegen können.

Mit der Gestaltung und Implementierung einer Web-basierten Benutzerschnittstelle, die alle Phasen des Quality-Improvement-Pradigmas (QIP) unterstützt, konnte der gegenwärtig verfügbare EDB-Prototyp [11, 27] bereitgestellt werden. Diese Schnittstelle implementiert spezifische, zum einen auf die einzelnen QIP-Phasen und zum anderen auf die verschiedenen, in SE-Prozessen auftretenden Rollen zugeschnittene Funktionen und Kontrollmechanismen. Für die Versionierung von Erfahrungsdaten wurde ein Konzept [13] erarbeitet, das in der kommenden Periode verfeinert und implementiert werden soll.

Basisframework für die Versionierung

Die Versionierung von Produktdaten ist eine Aufgabe, die in vielen Anwendungsbereichen gelöst werden muss. Trotz der Vielzahl von Versionierungssemantiken zur Unterstützung verschiedener Anwendungen mit unterschiedlichen Anforderungen gibt es einige grundlegende Eigenschaften der Versionierung. In diesem Punkt überlappt das bereits oben angesprochene SERUM-Projekt die Arbeitsschwerpunkte B und C. Dort soll ein Framework mit diesen grundlegenden Versionierungseigenschaften zur Verfügung gestellt werden, das an die anwendungsspezifische Versionierungssemantik angepasst und anschließend angewandt werden kann. Bisher wurden die grundlegenden Elemente für ein solches Framework herausgearbeitet [31, 31, 47]. Darüber hinaus wurde eine prinzipielle Vorgehensweise für die Anreicherung eines Produktdatenmodells um Versionierungsaspekte erarbeitet.

4 Zusammenarbeit mit anderen Teilprojekten

Neben der Teilnahme eines Mitarbeiters am Querschnittsprojekt Q1 [52] fanden in der Förderungsperiode 1998-2000 bereits zahlreiche Kooperationen mit anderen Teilprojekten statt, die in der kommenden Periode weiter gepflegt werden sollen.

Hervorzuheben ist hier die Kooperation mit dem Teilprojekt A1, die sich mit der Entwicklung des ORDBVS-basierten EDB-Prototyps beschäftigt. Eine Reihe von internen und externen Veröffentlichungen [10, 11, 27, 39] sowie Demonstrationen dokumentieren diese gute Zusammenarbeit. Als erstes Beispiel für die Anwendung der EDB bzw. für die Ablage von Erfahrungsdaten in der EDB wurden die im Teilprojekt B4 entwickelten SDL-Patterns verwendet [27, 39].

Wie bereits mehrfach angesprochen, wird eine Integration der Verwaltung von Erfahrungs- und Produktdaten im Rahmen der EDB angestrebt. Hier wurde das in Teilprojekt D1 entwickelte Produktdatenmodell beispielhaft betrachtet. Das Ergebnis dieser Kooperation ist in [12] beschrieben. Dieser erste Ansatz der Berücksichtigung eines Produktdatenmodells soll durch weitere Kooperation mit D1 verfeinert werden und darüber hinaus zu einem allgemeinen Ansatz zur Verwaltung (beliebig strukturierter) Produktdaten in der EDB ausgebaut werden.

In Ergänzung der bereits angesprochenen Kooperationen wurden erste Diskussionen mit den Teilprojekten A1, A2 und B2 geführt, die eine Integration von Erfahrungs-, Produktdatenverwaltung und Ablaufsteuerung zum Ziel haben. Dies bedeutet, dass die EDB durch Integration mit der Ablaufsteuerungskomponente MILOS zu einem umfassenden SE-Repository ausgestaltet werden soll. Hier wurden bereits erste Entwurfsüberlegungen angestellt. Diese sollen in der kommenden Periode ausgehend von einem in A3 erarbeiteten ersten Vorschlag [14] verstärkt fortgesetzt werden.

5 Eigene Veröffentlichungen im Berichtszeitraum

Diplomarbeiten

[1] Flehmig, M.: *Flexibilisierung der Verarbeitungskonzepte objekt-relationaler DBVS - Entwurfs- und Implementierungsaspekte*, Nov. 1998.

[2] Heinrich, J.: *Analyse von Konzepten zur Erweiterbarkeit objekt-relationaler Datenbanksysteme*, Feb. 1998.

[3] Gregor Johann: *Abbildung von OCL-Constraints auf SQL-Integritätsbedingungen*, Feb. 2000.

[4] Rieffel, T.: *Integration semantischer Beziehungen in ob-jekt-relationale Datenbanksysteme - Konzepte, Sprachgestaltung und Implementierungsanstäze*, Juni 1998.

[5] Rupp, A.: *Verfeinerung eines flexiblen Verarbeitungskonzepts im Hinblick auf ORDBMS-basierte Anwendungen*, März 2000.

[6] Sträter, M.: *Integration einer Ablaufsteuerungskomponente in ein objekt-relationales DBMS - Konzeption und Implementierungsaspekte*, Feb. 1999.

[7] Vorberg, A.: *Integration semantischer Beziehungen in objekt-relationale Datenbanksysteme - Benutzerschnittstelle, Sprachgenerierung und Precompiler*, Juni 1998.

[8] Andreas Weber: *GAURON - Gateway to UML-Repositories using Object-Relational Extensions*, Nov. 1999.

Dissertationen

[9] Zhang, N.: *Supporting Semantically Rich Relationships in Extensible Object-Relational Database Management Systems*, eingereicht, FB Informatik, 2000.

SFB-Berichte

[10] Feldmann, R.L., Mahnke, W., Ritter, N.:*(OR)DBMS-Support for the SFB 501 Experience Base*, Technical Report SFB 501-12/98, Fachbereich Informatik, Universität Kaiserslautern, 1998.

[11] Feldmann, R.L., Geppert, B., Mahnke, W., Ritter, N., Rößler, F.: *The ORDBMS-based SFB 501 Experience Base - Exemplified by the SDL-Pattern Approach,* SFB-Bericht 08/99, SFB 501, 1999.

[12] Mahnke, W., Marder, U., Queins, S., Ritter, N., Schuermann, B.: *Integrierte Produkt- und Erfahrungsdatenverwaltung*, SFB-Bericht 08/00, SFB 501, 2000, in Vorbereitung.

[13] Mahnke, W., Marder, U., Ritter, N.: *Versionierung in der Erfahrungsdatenverwaltung*, SFB 501, 2000, in Vorbereitung.

[14] Mahnke, W., Marder, U., Ritter, N.: *Ansätze der Integration von Erfahrungsdatenverwaltung und Ablaufsteuerung*, SFB 501, 2000, in Vorbereitung.

[15] Marder, U.: *Towards a Universal Media Server,* SFB-Report 03/2000, SFB 501, Feb. 2000.

[16] Marder, U.: *Introducing the VirtualMedia Concept,* Extended Abstract, 1999.

[17] Ritter, N., Steiert, H.-P.: *Exploiting Reuse Techniques in the SERUM Approach,* Interner Bericht, AG DBIS, Universität Kaiserslautern, 1999.

Vorträge

[18] Marder, U.: *Entwicklung und Einsatz einer Videodatenbank im WWW - ein Erfahrungsbericht*, Workshop "Multimedia-Systeme", Informatik ’98, 28. GI-Jahrestagung, Magdeburg, Sept. 1998.

[19] Härder, T.: *Object-Relational DBMSs - Type System, Modeling and Architectural Issues*, DB-Workshop "Extensibility and Component Architectures for Database Applications", München, März 1998.

[20] Härder, T.: *Generating Repository Managers for Cooperative Design Applications*, Data Engineering Forum, Kyoto, Japan, March 1999.

[21] Härder, T.: *An Integrated Information System for Technical Applications*, Computer Science Colloquium at Ochanomizu University and at University of Tokyo, Japan, March 1999.

[22] Ritter, N.: *Generating Call-Level Interfaces for Advanced Database Application Programming*, Informatik-Kolloquium der Univ. Stuttgart, Juni 1999.

[23] Ritter, N.: *Extensibility and Versioning in the Object-Relational Approach*, DB-Workshop "Extensibility and Component Architectures for Database Applications", München, März 1998.

[24] Steiert, H.-P.: *JPMQ - An Advanced Persistent Message Queueing Service*, DB-Workshop "Extensibility and Component Architectures for Database Applications", München, März 1998.

[25] Steiert, H.-P.: *Towards Generating Object-Relational Software Engineering Repositories,* Reuse-Workshop, SFB 501, Kaiserslautern, März 1999.

[26] Zhang, N.: *Enhanced Support of Relationship Semantics*, DB-Workshop "Extensibility and Component Architectures for Database Applications", München, März 1998.

Tagungsbeiträge

[27] Feldmann, R.L., Geppert, B., Mahnke, W., Ritter, N., Rößler, F.: *An ORDBMS-based Reuse Repository Supporting the Quality Improvement Paradigm - Exemplified by the SDL-Pattern Approach*, angenommen für TOOLS USA 2000, 34th Int. Conference & Exhibition, Santa Barbara, CA, July 2000.

[28] Gesmann, M., Härder, T.: *Supporting Parallel Navigation in Object-Relational Database Management Systems,* eingereicht zur Veröffentlichung, 2000.

[29] Härder, T., Mahnke, W., Ritter, N., Steiert, H.-P.: *Generating Repository Managers for Cooperative Design Applications,* in: Proc. 2nd Int. Symposium on Cooperative Database Systems for Advanced Applications (CODAS'99), Keynote-Paper, Springer, Singapore, 1999, 177-191.

[30] Loeser, H., Ritter, N.: *iWebDB - Integrated Web Content Management based on Object-Relational Database Technology,* in: Proc. Int. Database Engineering and Applications Symposium (IDEAS'99), Montreal, 1999, 92-97.

[31] Mahnke, W., Ritter, N., Steiert, H.-P.: *Towards Generating Object-Relational Software Engineering Repositories,* in: Proc. der GI-Fachtagung 'Datenbanksysteme in Büro, Technik und Wissenschaft' (BTW'99), Buchmann, A. (Hrsg.), Informatik aktuell, Springer-Verlag, Freiburg, März 1999, 251-270.

[32] Mahnke, W., Steiert, H.-P.: *Zum Einsatzpotential von ORDBVS in Entwurfsumgebungen,* in: Proc. Fachtagung CAD'2000, Berlin, März 2000, 219-239.

[33] Marder, U.: *Medienspezifische Datentypen für objekt-relationale DBMS: Abstraktionen und Konzepte,* in:Proc. der GI-Fachtagung 'Datenbanksysteme in Büro, Technik und Wissenschaft' (BTW'99), Buchmann, A. (Hrsg.), Informatik aktuell, Springer-Verlag, Freiburg, März 1999, 210-231.

[34] Marder, U.: *Improving the Performance of Media Servers Providing Physical Data Independence - Problems, Concepts, and Challenges,* in: Bertino, E., Heuer, A., Ozsu, T., Saake, G. (eds.): Multimedia Database Support for Digital Libraries, Dagstuhl-Seminar-Report 249, Schloss Dagstuhl, Sept. 1999.

[35] Marder, U.: VirtualMedia: *Making Multimedia Databases Fit for World-wide Access,* Ext. Abstract, EDBT'2000 Ph.D. Workshop, Konstanz, März 2000.

[36] Marder, U., Ritter, N., Steiert, H.-P.: *A DBMS-based Approach for Checking OCL Constraints,* OOPSLA'99-Workshop 'Rigourous Modeling and Analysis with the UML: Challenges and Limitations', Denver, Co., 1999.

[37] Nink, U., Härder, T., Ritter, N.: *Generating Call-Level Interfaces for Advanced Database Application Programming,* in: Proc. 25th Int. Conf. on Very Large Data Bases (VLDB'99), Edinburgh, Sept. 1999, 575-586.

[38] Ritter, N., Steiert, H.-P.: *Enforcing Modeling Guidelines in an ORDBMS-based UML Repository,* angenommen für International Resource Management Association Conference 2000 (Information Modeling Methods and Methodologies Track of IRMA 2000), Anchorage, Alaska, Mai 2000.

[39] Ritter, N., Steiert, H.-P., Mahnke, W., Feldmann, R.: *An Object-Relational SE-Repository with Generated Services,* in: Proc. International Resource Management Association Conference 1999, Managing Information Technology Resources in Organizations in the Next Millenium (Computer-Aided Software Engineering Track of IRMA'99), IDEA Group Publ., Mai 1999, 986-990.

[40] Steiert, H.-P.: *Towards a Component-based n-tier C/S-Architecture,* in: Proc. 3rd Int. Software Architecture Workshop (ISA W3), Orlando, Fl., ACM Press, 1998, 37-140.

[41] Surjanto, B., Ritter, N.: *XML Content Management based on Object-Relational Database Technology*, angenommen für 1st Int. Conf. on Web Information Systems (WISE 2000), Hong Kong, Juni 2000.

[42] Zhang, N., Härder, T.: *Enhanced Support of Relationship Semantics in Object-Relational Database Systems,* in: Proc. XIII Brazilian Symposium on Database Systems (SBBD'98), Maringa, Okt. 1998, 7-21.

[43] Zhang, N., Härder, T.: *On a Buzzword "Extensibility" - What we have learned from the ORIENT Project,* in: Proc. Int. Database Engineering and Applications Symposium (IDEAS'99), Montreal, Canada, 1999, 360-369.

[44] Zhang, W.P., Ritter, N.: *Measuring the Contributions of (O)RDBMS to Object-Oriented Software Development*, angenommen für Int. Database Engineering and Applications Symposium (IDEAS 2000), Yokohama, Japan, Sept. 2000.

[45] Zhang, W.P., Ritter, N.: *Leistungsuntersuchungen hinsichtlich der Nutzung von (O)RDBVS als persistente Objektsysteme,* 12. GI-Workshop "Grundlagen von Datenbanken, Holsteinische Schweiz, Juni 2000.

Zeitschriften / Monografien

[46] Avenhaus, J., Gotzhein, R., Härder, T., Litz, L., Madlener, K., Nehmer, J., Richter, M., Ritter, N., Rombach, H.D., Schürmann, B., Zimmermann, G.: *Entwicklung großer Systeme mit generischen Methoden - Eine Übersicht über den Sonderforschungsbereich 501*, Informatik - Forschung und Entwicklung, 13(4), 1998, 227–234.

[47] Härder, T., Mahnke, W., Ritter, N., Steiert, H.-P.: *Generating Versioning Facilities for a Design Data Repository Supporting Cooperative Applications,* Int. Journal on Intelligent & Cooperative Inf. Systems 9:1-2, 2000, 117-146.

[48] Härder, T., Nink, U., Ritter, N: *Generierte DB-Aufrufschnittstellen - Anwendungsspezifische Zugriffsoptimierung durch Bindungsflexibilität,* Informatik - Forschung und Entwicklung 15(2), Springer-Verlag, 2000.

[49] Käfer, W., Ritter, N., Schöning, H.: *Konfigurierungskonzepte für datenbankbasierte, technische Entwurfsanwendungen,* Informatik - Forschung und Entwicklung 13(1), Springer-Verlag, 1998, 1-17.

6 Fremdveröffentlichungen

[50] Feldmann, R.L.: *Developing a Repository Structure that Allows Comprehensive Reuse and Continuous Improvement*, in: Proc. 8. Kolloquium Software-Entwicklung – Methoden, Werkzeuge, Erfahrungen, Ostfildern, Technische Akademie Esslingen, Sept. 1999.

[51] Franklin, M.J., Carey, M., Livny, M.: *Transactional Client-Server Cache Consistency: Alternatives and Performance*, ACM TODS 22:3, 1997, 315-363.

[52] Queins, S., Zimmermann, G.: *A First Iteration of a Reuse-Driven, Domain-Specific System Requirements Analysis Process*, Technical Report SFB 501-13/99, Fachbereich Informatik, Universität Kaiserslautern, 1999.

[53] Stonebraker, M., Brown, P., Moore, D.: *Object-relational DBMSs - The Next Wave*, Second Edition, Morgan Kaufmann, 1998.

[54] SQL99: *ANSI/ISO/IEC 9075-1-1999 Database Languages SQL Part 1 Framework.*

[55] SQL99: *ANSI/ISO/IEC 9075-2-1999 Database Languages SQL Part 2 Foundation.*

Teilprojekt B1

Generische Modellierung von Prozessen und Experimenten

Leiter

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

Mitarbeiter

Ergänzungsausstattung:

Dipl.-Inform. Jürgen Münch

Grundausstattung:

Dipl.-Inform. Marcus Ciolkowski (seit 15.7.99)
Dipl.-Inform. Christiane Differding (bis 30.6.99)
Dipl.-Inform. Karin Drabe (bis 15.08.98)
Prof. Dr. Victor Basili (externer Berater)

1 Kurzfassung der Aufgabenstellung

Langfristiges Ziel dieses Teilprojekts ist die Entwicklung einer generisch ausgelegten Experimentiermethodik für die Gewinnung detaillierter Erfahrungswerte bezüglich der Eignung von Entwicklungstechniken und -prozessen in Bezug auf die relevanten Projektziele in der Domäne Gebäudeautomation unter Berücksichtigung der gegebenen Randbedingungen. Dies soll durch leichte Instantiierung generischer Experimentpläne bzw. generischer Prozessmodelle unterstützt werden. Die Aufgabenstellung umfasst die wissenschaftliche Auseinandersetzung mit folgenden Themenbereichen:

a) Entwicklung einer werkzeuggestützten Methodik zur Generierung von Messplänen, die die Spezifikation und Analyse experimenteller Messziele unterstützen;
b) Untersuchung, Realisierung und Erprobung von Generierungstechniken zur effektiven und effizienten Bereitstellung maßgeschneiderter Prozessmodelle;
c) Entwicklung einer Methodik zur Bereitstellung maßgeschneiderter, umfassender Projektpläne, die verschiedene Sichten (z. B.: Messsicht, Prozesssicht) konsistent integrieren;
d) Definition eines Schemas zur Ablage experimentell gewonnener Erfahrungen in einer Erfahrungsdatenbank.

In der ersten Förderungsperiode wurden Methoden, Techniken und Prototypwerkzeuge zur Beschreibung von Messplänen und Prozessmodellen sowie Schemata für die Ablage experimentspezifischer Ergebnisse bereitgestellt.

In der zweiten Förderungsperiode sollten diese Methoden, Techniken und Prototypwerkzeuge erprobt und optimiert werden. Die Optimierung sollte einerseits hinsichtlich der Erhöhung der Wiederverwendbarkeit von Messplänen und Prozessmodellen durch deren generische Auslegung für das Anwendungsgebiet „Gebäudeautomation" erfolgen. Andererseits sollte die Wiederverwendbarkeit von Ergebnissen vergangener Experimente durch deren Ablage im experimentübergreifenden Datenbereich der Erfahrungsdatenbank des SFB erhöht werden.

2 Entwicklung des Teilprojekts in der Förderperiode

Auf die im Berichtszeitraum erzielten Ergebnisse wird in Abschnitt 3 eingegangen.

Im Berichtszeitraum wurden zwei Dissertationen verfasst:

In Rahmen der von C. Differding verfassten Dissertation *Supporting Development, Modification, and Reuse of Measurement Plans* [17] wurde eine Methodik zur Erstellung und Wartung von Messplänen unter Wiederverwendung existierender Projektinformationen (insbesondere Prozess- und Qualitätsmodelle) entwickelt. Diese Methodik ist werkzeuggestützt. Des Weiteren wurde ein Algorithmus zur Verfeinerung von Messplänen aus Messzielen entwickelt. Dieser Algorithmus ermöglicht eine

Teilautomatisierung des Verfeinerungsprozesses von Messplänen und bietet somit wesentliche Unterstützung bei der Generierung von Messplänen basierend auf Projektinformation.

Im Rahmen der von J. Münch verfassten Dissertation *Zielführende, wiederverwendungsorientierte Erstellung von Software-Projektplänen* [18] wurde eine werkzeuggestützte Planungsmethodik zur Erstellung von Software-Projektplänen entwickelt. Der in dieser Arbeit entwickelte Ansatz ist umfassend in dem Sinne, dass er einen Planungsprozess ausgehend von einer Definition der Ziele und der Charakterisierung eines Software-Entwicklungsprojekts bis zur Erstellung eines maßgeschneiderten Projektplans beschreibt. Die entwickelte Methodik ist auf die Vorgehensplanung beschränkt, d. h. die Beschreibung der technischen Einzelprozesse und deren Zusammenspiel. Techniken, die im Rahmen dieses Ansatzes neu entwickelt wurden, betreffen a) die Komposition wiederverwendbarer Artefakte, die Prozesswissen beinhalten, b) die Verifikation von Plänen bzw. Teilplänen gegen formal definierte Ziele sowie c) die projektspezifische Anpassung von Prozessmodellen, die in einer Prozessmodellierungssprache formuliert sind. Der Ansatz basiert auf einem im Rahmen dieser Arbeit entwickelten konzeptuellen Modell für Software-Entwicklungsprozesse.

Die Ergebnisse des Teilprojekts haben von der Kooperation mit und von der begleitenden Begutachtung durch externe Experten aus dem Fraunhofer-Institut für Experimentelles Software Engineering IESE sowie aus dem Ausland (insbesondere Prof. Basili von der University of Maryland und Prof. Iida, Nara Institute of Science and Technology, Japan) profitiert. Prof. Basili hat sich mehrere Male zu Projektbegutachtungen in Kaiserslautern aufgehalten. Prof. Iida erarbeitete im Rahmen eines halbjährigen Forschungsaufenthalts (1. Juni 1998 bis 30. November 1998) einen Ansatz zur Kapselung von Wissen über Software-Entwicklungsaktivitäten mittels Prozessmustern. Hierbei identifizierte er basierend auf im SFB angewandten Vorgehensmodellen Prozessmuster und entwickelte eine Vorgehensweise zur Kombination dieser Muster [57]. Über die Kooperation mit dem Fraunhofer-Institut IESE konnten die entwickelten Techniken und Methoden unter industriellen Randbedingungen (z. B.: bei der Allianz Lebensversicherungs AG, bei der Bosch Telecom GmbH [jetzt Tenovis]) erprobt werden.

Im Berichtszeitraum fand in Kaiserslautern die internationale Konferenz SEKE'99 (11th International Conference on Software Engineering and Knowledge Engineering) statt. Der Teilprojektleiter war General Chair der Konferenz.

Im Berichtszeitraum wurde der 6. Workshop der GI-Fachgruppe 5.1.1 zum Thema "Vorgehensmodelle, Prozessverbesserung und Qualitätsmanagement" durchgeführt (in Kooperation mit dem Fraunhofer-Institut IESE).

Die Ergebnisse des Teilprojekts wurden bereits weltweit veröffentlicht (z. B.: International Journal of Empirical Software Engineering, International Conference on Software Engineering and Knowledge Engineering) und sind auf positive Resonanz in Fachkreisen gestoßen.

3 Durchgeführte Arbeiten und Ergebnisse

3.1 Arbeitsschwerpunkt A: Generische Modellierung von GQM-Plänen

Identifikation relevanter Parameter

Ziel dieses Arbeitspakets war die Identifikation von Parametern, die Struktur und Inhalt von GQM-Plänen wesentlich beeinflussen, sowie die Untersuchung der Wirkung der identifizierten Parameter auf GQM-Pläne. In der zweiten Förderungsperiode wurde der Einfluss ausgewählter GQM-Zielparameter auf GQM-Pläne untersucht. Ziele von GQM-Plänen variieren in Abhängigkeit von Objekt und Zweck der Messungen, vom gewählten Qualitätsaspekt, von der Sicht sowie vom Kontext. Speziell untersucht wurden Variationen der Parameter Objekt und Qualitätsmerkmal. Der Parameter Objekt beschreibt Informationen zur Beantwortung der Frage: Was wird gemessen? Sei beispielsweise ein Vorgehensmodell das Objekt in einem Messprogramm zur Charakterisierung des Aufwandes. Die Hinzufügung eines Prozessschrittes führt dann im GQM-Plan zu einer Hinzufügung einer entsprechenden GQM-Frage bzw. eines entsprechenden Maßes. Der Parameter Qualitätsmerkmal beschreibt Informationen zur Beantwortung der Frage: Welche Qualitätseigenschaft wird gemessen? Sei beispielsweise der Defect Slippage (d. h., die Verteilung der Fehler hinsichtlich ihres Entdeckungszeitpunkts) das zu untersuchende Qualitätsmerkmal und man verändere die Fehlerklassifikation. Die Wirkung einer solchen Parameteränderung auf den GQM-Plan besteht in der Veränderung des unmittelbar zugehörigen Maßes sowie der Anpassung hiervon abhängiger Maße. Des Weiteren müssen unter Umständen neue mit GQM-Fragen assoziierte Berechnungsvorschriften zur Interpretation der Messwerte in Bezug auf den Defect Slippage in den GQM-Plan eingefügt werden. Der Parameter Zweck beschreibt Informationen zur Beantwortung der Frage: Wozu wird gemessen? Im Rahmen der in diesem Arbeitspaket durchgeführten detaillierten Untersuchung der Parameter wurde der Parameter Zweck zunächst auf den Zweck des Charakterisierens fixiert. Dies ist unter anderem damit begründet, dass bei den im SFB durchgeführten Baselining-Entwicklungen eine Charakterisierung wichtiger Qualitätsmerkmale im Vordergrund steht. Die beiden variierten Parameter Objekt und Qualitätsmerkmal werden repräsentiert durch Prozessmodelle beziehungsweise Qualitätsmodelle. Der Einfluss dieser Parameter auf GQM-Pläne wurde bei der Entwicklung der Methodik zur Erstellung und Wartung von Messplänen [17] maßgeblich berücksichtigt.

Definition einer Notation zur generischen Formulierung von GQM-Plänen

Ziel dieses Arbeitspakets war die Entwicklung einer Notation, die generische GQM-Pläne beschreibt. In der zweiten Förderungsperiode wurde ein formales Modell für die Darstellung von GQM-Plänen entwickelt [17]. Dieses Modell wurde mit einem (zusätzlich entwickelten) Modell für Prozess- und Qualitätsmodelle integriert.

Entwicklung einer Methodik zur generischen Erstellung und Wartung von GQM-Plänen

Ziel dieses Arbeitspakets war die Entwicklung einer Methodik zur Erstellung und Wartung von generischen GQM-Plänen. Insbesondere sollten Vorgehensweisen und Richtlinien bereitgestellt werden, die die schrittweise Erstellung von GQM-Plänen unter Ausnutzung wiederverwendbaren Erfahrungswissens sowie die Durchführung von Änderungen an GQM-Plänen unterstützen. In der zweiten Förderungsperiode wurde ein Erstellungs- und Wartungsprozess für Messpläne (die GQM-Pläne enthalten) entwickelt [17]. Dieser Prozess enthält unter anderem einen auf dem integrierten Modell basierenden Algorithmus zur Verfeinerung von GQM-Plänen und bildet damit die Grundlage zur Generierung von GQM-Plänen aus Messzielen, Prozess- und Qualitätsmodellen.

Realisierung von Werkzeugen zur Erstellung und Wartung von GQM-Plänen

Ziel dieses Arbeitspakets war die Entwicklung einer Werkzeugungerstützung zur Bearbeitung von GQM-Plänen. In der zweiten Förderungsperiode wurde ein Werkzeug erstellt [6] und weiterentwickelt [2][4], mit dem GQM-Pläne erstellt und exportiert werden können. Zur Unterstützung der Wartung von GQM-Plänen wurde das Werkzeug erweitert um eine Schnittstelle zum Einlesen von geänderten Prozessmodellen [4]. Basierend auf dieser Information wird die Wartung von GQM-Plänen bei Änderungen des Prozessmodells unterstützt [2].

3.2 Arbeitsschwerpunkt B: Generische Modellierung von Prozessmodellen

Identifikation relevanter Parameter

Ziel dieses Arbeitspakets war die Identifikation geeigneter Parameter für Software-Entwicklungsprozesse des Anwendungsgebiets “Gebäudeautomation”. In der zweiten Förderungsperiode wurden existierende Methoden, die zur Entwicklung eingebetteter Software geeignet oder bestimmt sind, sowie ausgewählte Varianten dieser Methoden in der formalen Prozessmodellierungssprache MVP-L formuliert. Eine vergleichende Analyse der Varianten führte zur Identifikation wesentlicher Einflussfaktoren und einer Entwicklung konkreter Charakterisierungsschemata. Im Einzelnen wurden folgende Methoden modelliert (die Beschreibung der Modellierung findet sich in den angegebenen Referenzen): 1) SLD-oriented Object Modeling Technique (SOMT) [19], 2) Naval Research Laboratory / Software Cost Reduction (NRL / SCR) [12], 3) Object Modeling Technique (OMT) [12], 4) Der Pss-05-Prozess der European Space Agency [5]. Wesentliche Einflussfaktoren wie Echtzeitfähigkeit und Sicherheitskritikalität sowie ermittelte Charakterisierungsschemata (starr, facetten- und modellorientiert) sind in [5] und [12] beschrieben.

Definition einer Notation zur Formulierung generischer Prozessmodelle

Ziel dieses Arbeitspakets war die Entwicklung einer Notation zur Formulierung generischer Prozessmodelle. In der zweiten Förderungsperiode wurde ein konzeptuelles Modell entwickelt, dass die wesentlichen Konzepte von Software-Entwicklungsprozessen umfasst und deren Semantik mittels Zusicherungen definiert [8]. Das Modell orientiert sich am Vier-Schichten-Modell zur Metamodellierung von Information (Notationsschicht, Konzeptschicht, Modellschicht und Projektschicht) und ist im Wesentlichen auf der Konzeptschicht angesiedelt. Das konzeptuelle Modell ist spezifiziert unter Verwendung folgender Beschreibungstechniken: a) Unified Modeling Language (UML, Version 1.3). Hierbei werden Klassendiagramme zur Beschreibung des konzeptuellen Modells verwendet. Des Weiteren wird das UML-Metamodell (meta-metamodel, metamodel, model, user objects) zur Einordnung der Modellelemente benutzt; b) Object Constraint Language (OCL, Version 1.3). OCL wird hierbei benutzt zur Beschreibung von Constraints von UML-Modellen. OCL wurde zur Beschreibung der Constraints des konzeptuellen Modells um benutzerdefinierte Funktionen erweitert. Das konzeptuelle Modell ist konform mit den Konzepten der Prozessmodellierungssprache MVP-L, umfasst jedoch Konstrukte (z. B.: charakterisierende Attribute, Gültigkeitsbereich, Anpassungsregeln) zur Darstellung von Generik in Prozessmodellen.

Entwicklung einer Methodik zur Erstellung maßgeschneiderter Prozessvarianten in MVP-L

Ziel dieses Arbeitspakets war die Entwicklung einer Methodik zur effektiven und effizienten Bereitstellung maßgeschneiderter Prozessmodelle in MVP-L. In der zweiten Förderungsperiode wurde basierend auf dem konzeptuellen Modell ein werkzeuggestützter Ansatz zur Erstellung von maßgeschneiderten Prozessmodellen entwickelt. Dieser Ansatz ist integriert in eine zielführende, wiederverwendungsorientierte Planungsmethodik zur Erstellung von Projektplänen [18]. Techniken, die im Rahmen dieses Ansatzes neu entwickelt wurden, betreffen a) die Komposition wiederverwendbarer Artefakte, die Prozesswissen beinhalten (in Form von Prozessmustern [57]), b) die Verifikation von Plänen bzw. Teilplänen gegen formal definierte Ziele (in Analogie zur Verifikation von Programmen nach [59]) sowie c) die projektspezifische Anpassung von Prozessmodellen, die in MVP-L formuliert sind [46]. Die Methodik ist fünfschrittig (Selektion, Komposition, Evaluierung, Anpassung, Integration) und kann iterativ durchgeführt werden.

Implementierung von Werkzeugen zum Maßschneidern von Prozessmodellen

Ziel dieses Arbeitspakets war die Realisierung einer prototypischen Werkzeugunterstützung zur (Teil-)Automatisierung von notwendigen Generierungsschritten bei der Erstellung maßgeschneiderter Prozessmodelle. In der zweiten Förderungsperiode wurde ein unterstützendes Werkzeug entwickelt, dass die Anpassung von Prozessmodellen an Projektziele und -charakteristika weitgehend automatisiert. Dieses Werkeug, das sogenannte Process Tailoring Tool (ProTail), basiert auf einem transformationsbasierten Ansatz. Die Anpassung der Prozessmodelle erfolgt hierbei durch eine Folge von Transformationsschritten, die mit der sequentiellen Anwendung von Anpassungsregeln und Anpassungsprimitiva (hier in Form von Transformationsregeln) verglichen werden können. Der transformationsbasierte Ansatz bietet die Möglichkeit, Anpassungsregeln explizit zu erfassen, sodass sie einfach an neue Erkenntnisse angepasst werden können. Der Anpassungsprozess wird hierbei unterteilt in die Teilschritte

Konfigurierung, Charakterisierung, Transformation und Dokumentation. Abb. 1 gibt einen Überblick über die Eingangs- und Ausgangsdokumente des Werkzeugs. Das Werkzeug enthält zusätzlich eine Komponente zur Verwaltung von Qualitätsmodellen, sodass bekannte Relationen zwischen relevanten Qualitätsmerkmalen und zugehörigen Einflussfaktoren in den Anpassungsprozess einbezogen werden können [13].

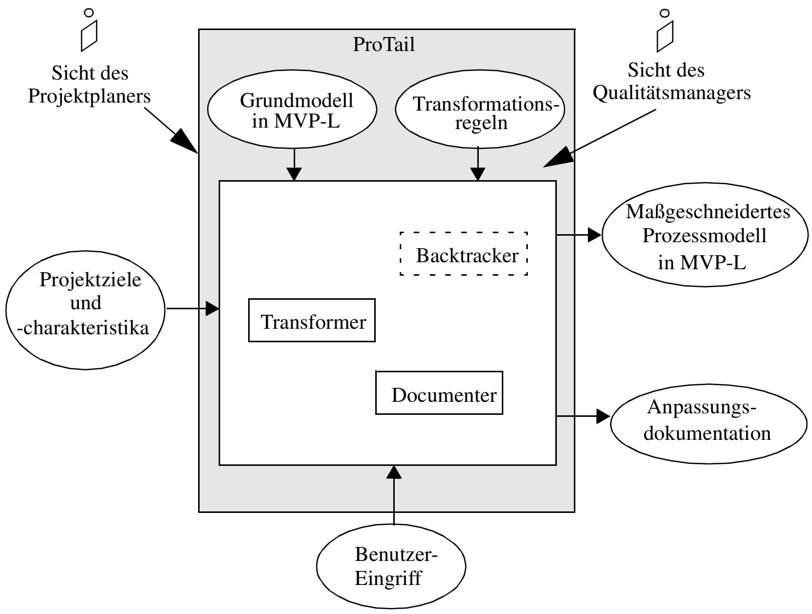

Abbildung 1: Werkzeuggestützte Anpassung von Projektmodellen

3.3 Arbeitsschwerpunkt C: Generierung instrumentierter Projektpläne

Ziel dieses Arbeitsschwerpunkts war die Entwicklung einer systematischen Methodik zur Erstellung umfassender Projektpläne, die die Mess- und Projektsicht integrieren. Die in der zweiten Förderungsperiode entwickelte Methodik zur generischen Modellierung von GQM-Plänen [17] und das zugehörige Werkzeug GQMPlanner [4] ermöglichen die Entwicklung von GQM-Plänen, die konsistent zu einem existierenden Prozessmodell sind. Dies wird durch die Erstellungs- und Wartungsmethode gesichert. Somit können Prozessmodelle und GQM-Pläne separat und koordiniert erstellt werden, müssen jedoch anschließend integriert werden. Eine solche Vorgehensweise hat zunächst den Vorteil, dass Prozess- und Messaspekte separat untersucht und Variationen innerhalb der jeweiligen Sicht identifiziert werden können. Es ergibt sich jedoch das Problem der 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. 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. Um dieses

Problem zu überwinden, soll die Methodik in der dritten Förderungsperiode dahingehend weiterentwickelt werden, dass Erstellungsprozesse für Mess- und Prozesspläne 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.

3.4 Arbeitsschwerpunkt D: Organisation der Erfahrungsdatenbank

Ziel dieses Arbeitsschwerpunkts war die Entwicklung eines experimentübergreifenden Erfahrungsdatenbankschemas basierend auf einer vergleichenden Analyse der im SE-Kern des SFB existierenden Erfahrungen. In der zweiten Förderungsperiode wurde in enger Kooperation mit Teilprojekt A1 die existierenden Erfahrungen im SE-Kern analysiert und ein grobgranulares Datenbankschema für den experimentübergreifenden Bereich der Erfahrungsdatenbank ermittelt [44]. Dieses Schema wurde von Teilprojekt A1 verfeinert und in Zusammenarbeit mit Teilprojekt A3 realisiert.

3.5 Arbeistschwerpunkt E: Evaluierung entwickelter Techniken und Werkzeuge

Ziel dieses Arbeitsschwerpunkts war eine Evaluierung der entwickelten Methoden, Techniken und Werkzeuge. Dazu sollte insbesondere das SFB-Praktikum herangezogen werden. In der zweiten Förderungsperiode wurde die Methodik und das Werkzeug zur Erstellung von GQM-Plänen in einer initialen Anwendung [1] und weiter in einer Serie von Fallstudien evaluiert [9]. Die Effektivität und Effizienz des Prozesses zur Wartung von GQM-Plänen wurde in einem kontrollierten Experiment mit Studenten untersucht. Dabei wurde das Prozessmodell verändert, und der GQM-Plan wurde von den Studenten angepasst. Die Wartung wurde in diesem Experiment auf den initialen Schritt der Auswirkungsanalyse reduziert. Es wurde gezeigt, dass die Wartung durch ein integriertes Modell von GQM-Plan und Prozessmodell effektiver (höhere Korrektheit der identifizierten Auswirkungen) und effizienter (weniger Aufwand zum Identifizieren der Auswirkungen) ist als die Wartung ohne ein solch integriertes Modell. Außerdem ist den Studenten die Wartung mit dem integrierten Modell als Basis subjektiv leichter gefallen. Die Methodik und das Werkzeug zur Erstellung maßgeschneiderter Prozessmodelle wurden in zwei Fallstudien [18] hinsichtlich der erwarteten Reduktion des Erstellungsaufwandes und der erwarteten Erhöhung der Konsistenz der erstellten Prozessmodelle untersucht. Dabei konnte die Zeit, die zur werkzeuggestützten Generierung eines Prozessmodells benötigt wurde, gegenüber einer manuellen Erstellung signifikant reduziert werden (Faktor >65). Komplett eliminiert werden konnten Zeiten zur Identifizierung von Folgeänderungen und zur Erstellung der Dokumentation, da dies automatisch durch das Werkzeug geschieht. Sowohl bei der manuellen Erstellung als auch bei der automatischen Generierung maßgeschneiderter Prozessmodelle konnte die Konsistenz durch den Einsatz eines automatischen Konsistenzcheckers gewährleistet werden. Hierbei ist jedoch zu beachten, dass bei der manuellen Erstellung das Prozessmodell solange auf Inkonsistenzen geprüft wurde, bis das Ergebnis konsistent wurde. Bei der werkzeuggestützten Generierung war nur eine Konsistenzprüfung am Ende des Generierungsvorgangs nötig. Weiterhin ist zu bemerken, dass diese Verwendung eines Konsistenzcheckers zur Erleichterung der manuellen Erstellung nur deshalb möglich wurde, da auch die manuelle Erstellung auf der Ebene von MVP-L, d. h. einer formalen Prozessmodellierungssprache, stattfand. Bei der Verwendung einer informellen Beschreibung in Form von Textbausteinen wäre dies nicht möglich gewesen. Es ist zu vermuten, dass ohne die Verwendung eines Konsistenzcheckers (also einer tatsächlich 100%-igen manuellen Anpassung) Inkonsistenzen wesentlich häufiger auftreten würden, da Folgeänderungen übersehen würden. Abschließend sei bemerkt, dass die in Teilprojekt B1 entwikkelten Techniken, Methoden und Werkzeuge zur Unterstützung der Querschnittsprojekte Q1 und Q2 eingesetzt wurden und hierbei gemachte Erfahrungen bei der Weiterentwicklung berücksichtigt wurden. Querschnittsprojekt Q2 wurde im Rahmen des SFB-Praktikums 1999 durchgeführt.

4 Zusammenarbeit mit anderen Teilprojekten

Die Durchführung von Anwendungsentwicklungen und Experimenten erfolgte unter Anleitung des SE-Labors (Teilprojekt A1). Hierfür wurden die Ergebnisse des Teilprojekts B1 in Form von Techniken und Prototypwerkzeugen in das SE-Labor integriert und damit dem gesamten SFB zur Verfügung gestellt. Zusätzlich wurde in Zusammenarbeit mit Teilprojekt A1 das grobgranulare Schema des experimentübergreifenden Teils der Erfahrungsdatenbank definiert und von A1 und A3 realisiert.

In Zusammenarbeit mit den Teilprojekten A2 und B2 wurde der MILOS-Ansatz, der MVP-L mit plan-basierten KI-Techniken integriert, um Konzepte der verteilten Planung erweitert [50][52][55].

In Zusammenarbeit mit Teilprojekt B4 wurde die SOMT-Entwicklungsmethode formal modelliert [19].

Der Mitarbeiter der Ergänzungsausstattung arbeitete im Querschnittsprojekt Q1 bei der Mess-und Projektplanung mit [21].

Das Querschnittsprojekt Q2 (SFB-Praktikum im Sommersemester 1999) wurde von Mitarbeitern des Teilprojekts mitbetreut [23][24].

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] Eleni Chaitidou: *Erstellung von Projekt- und GQM-Plänen für ein Software-Entwicklungsprojekt.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.

[2] Matthias Gutheil: *Wartbare Schema-Repräsentation für GQM-Pläne.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.

[3] Sascha Werner: *Entwurf und Realisierung einer ergonomischen Benutzungsschnittstelle.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.

[4] Eric Ras: *Export-/ Importschnittstelle für das Werkzeug GQMPlanner.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[5] Björn Schmidt: *Modellierung eines adaptierbaren Software-Entwicklungsprozesses.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[6] Christian Voigtländer: *GQMaspect II: Entwicklung eines Werkzeugs zur Erstellung von GQM-Plänen.* Projektarbeit, Fachbereich Informatik (mitbetreut durch Fraunhofer IESE), Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[7] Wolfgang Wagenbichler: *Erstellung einer SW-Systemdokumentation mit StP.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[8] Jens Heidrich: *Conceptual Modeling of Software Development Processes.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2000.

[9] Jürgen Kortgen: *Validierung der systematischen Entwicklung von GQM-Plänen.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2000.

Diplomarbeiten

[10] Kerstin Brockhage: *Suche nach einem Verfolgbarkeitsansatz zur Verbesserung von Änderbarkeit (Search for a traceability approach to improve changeability).* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.

[11] Ralph Eiselein: *Definition of measurement points and measurement procedures by using process models and GQM plans.* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.

[12] 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.

[13] Thomas Ruf: *Erweiterung eines Werkzeugs zur interaktiven Manipulation von Software-Projektplänen (Extension of a tool for the interactive manipulation of software project plans).* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1998.

[14] Marcus Ciolkowski: *Evaluating the Effectiveness of Different Inspection Techniques on Informal Requirements Documents.* Diplomarbeit, Fachbereich Informatik (mitbetreut durch Fraunhofer IESE), Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[15] Marco Heide: *Objektorientierter Entwurfsprozess zur Gewährleistung von Verfolgbarkeit im Bereich technischer Systeme.* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[16] Christian Voigtländer: *Werkzeugunterstützung zur Aufbereitung von Auswertungen im Rahmen von Messprogrammen.* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 2000.

Dissertationen

[17] Christiane Differding: *Supporting Development, Modification, and Reuse of Measurement Plans*. Dissertation (zur Einreichung im Berichtszeitraum vorgesehen), Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 2000.

[18] Jürgen Münch: *Zielführende, wiederverwendungsorientierte Erstellung von Software-Projektplänen.* Dissertation (zur Einreichung im Berichtszeitraum vorgesehen), Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 2000.

SFB-Berichte

[19] 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.

[20] Kerstin Brockhage, Christiane Differding und Antje von Knethen: *Bessere Änderbarkeit durch Verfolgbarkeit: Auswahl eines Ansatzes.* Technischer Bericht Nr. 301/99, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[21] Raimund L. Feldmann, Jürgen Münch, Stefan Queins, Stefan Vorwieger und Gerhard Zimmermann: *Baselining a Domain-Specific Software Development Process.* Technischer Bericht Nr. 02/97, Sonderforschungsbereich 501, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[22] Jürgen Münch: *Reuse of Process Patterns.* In Tagungsband des Workshops "Software Reuse", Technischer Bericht Nr. 5/99, Sonderforschungsbereich 501, Fachbereich Informatik,Universität Kaiserslautern, 67653 Kaiserslautern, 22.-23. Februar 1999.

[23] 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.

[24] 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, Universitt Kaiserslautern, 67653 Kaiserslautern, 2000.

Vorträge

[25] H. D. Rombach: *Continuous Improvement of Software Development Competence: Prerequisite for Future Competitiveness*, Espoo & Oulu, Februar 1998.

[26] H. D. Rombach: *Kontinuierliche Qualitätsverbesserung*, ESA, Amsterdam, Niederlande, März 1998.

[27] H. D. Rombach: *Software Experience Center Project*, Ericsson, Irland, März 1998.

[28] H. D. Rombach: *Software-Prozessverbesserung: Strategische Notwendigkeit für alle Branchen*, SQM-Tagung, 1998, Köln, April 1998.

[29] H. D. Rombach: *Experimentation as a Vechicle for Software Technology Transfer: A Family of Software Reading Techniques*, „FESMA '98", Konferenz, (Keynote), Antwerpen, Mai 1998.

[30] H. D. Rombach: *Die Bedeutung von Experimenten im Software Engineering*, Kolloquium, Graduiertenkolleg „Parallele & Verteilte Systeme", Universität Stuttgart, Juni 1998.

[31] H. D. Rombach: *Experimentieren: Prototyping bei der Prozessverbesserung*, GI-Konferenz „Softwaretechnik '98", Paderborn, August 1998.

[32] H. D. Rombach: *Softwaretechnologie: Entscheidender Faktor für Wirtschaftswachstum*, Jahrestagung der FhG, Darmstadt, Oktober 1998.

[33] H. D. Rombach: *Empirische Methoden: Voraussetzung für ingenieurmäßige Softwareentwicklung*, Eingeladener Festvortrag, 10-Jahresfeier Fakultät für Informatik, Universität Stuttgart, November 1998.

[34] H. D. Rombach: *Testing and Inspections: A Historical Perspective*, „EuroStar98“, Konferenz, (Keynote) München, Dezember 1998.

[35] H. D. Rombach: *Software Experience Factory: Basis für beschleunigte Prozessverbesserung*, GI-Workshop, Kaiserslautern, April 1999.

[36] H. D. Rombach: *Innovationen im Software-Entwicklungsprozess*, Robert-Bosch Kolleg, Stuttgart, Juni 1999.

[37] H. D. Rombach: *Software Process Improvement: Prerequisite for Future Competitiveness*, SEC Meeting, Baden, Schweiz, September 1999.

[38] H. D. Rombach: *Continuous Improvement of Software Development Competence: Prerequisite for Future Competitiveness*, University of New South Wales, New South Wales, Australia, November 1999.

[39] H. D. Rombach: *Experimental Software Engineering*, Colloquium, Nara Institute of Science and Technology (NAIST), School of Information Science, Nara, Japan, November 1999.

[40] H. D. Rombach: *Experimental Software Engineering*, Colloquium, Osaka University, Graduate School of Engineering Science, Department of Computer Science & Mathematics, Osaka, Japan, November 1999.

[41] H. D. Rombach: *GERMANY: Activities in Software Process Improvement, Country Report for Germany* (invited presentation), Indian SEPG Conference, Bangalore, India, Februar 2000.

[42] H. D. Rombach: *Capitalising on Experience – What is missing in CMM & PSP/TSP?* -, (keynote address), Indian SEPG Conference, Bangalore, India, Februar 2000.

[43] H. D. Rombach: *Software Experience Factory „how to do it“* Guide, (tutorial), Indian SEPG Conference, Bangalore, India, Februar 2000.

Tagungsbeiträge

[44] Raimund L. Feldmann, Jürgen Münch und Stefan Vorwieger: *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.

[45] 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.

[46] 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.

[47] Sigrid Goldmann, Jürgen Münch und Harald Holz: *MILOS: A Model of Interleaved Planning, Scheduling, and Enactment.* In Web-Proceedings of the 2nd Workshop on Software Engineering over the Internet, Los Angeles, CA, USA, 17. Mai 1999.

[48] 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.

[49] 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.

[50] 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.

[51] Dieter Rombach: *Experimentation: Engine for Applied Research and Technology Transfer in Software Engineering.* In Proceedings of the 24th Annual Software Engineering Workshop of the SEL, Greenbelt, USA, 1./2. Dezember 1999.

[52] 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.

Zeitschriften / Monografien

[53] 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.

[54] J. Avenhaus, R. Gotzhein, T. Härder, L. Litz, K. Madlener, J. Nehmer, M. Richter, N. Ritter, H. D. Rombach, B. Schürmann und G. Zimmermann: *Entwicklung großer Systeme mit generischen Methoden - Eine Übersicht über*

den Sonderforschungsbereich 501. Informatik, Forschung und Entwicklung, 13(4):227–234, Dezember 1998.

[55] Sigrid Goldmann, Jürgen Münch und Harald Holz. *Distributed Process Planning Support with MILOS.* In International Journal of Software Engineering and Knowledge Engineering, August 2000.

6 Fremdveröffentlichungen

[56] V.R. Basili, G. Caldiera und H.D. Rombach: *Goal Question Metric Paradigm*; In John J. Marciniak, Herausgeber, Encyclopedia of Software Engineering, Band 1, Seiten 528-532, John Wiley & Sons, 1994.

[57] 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.

[58] S. Queins und G. Zimmermann: *A First Iteration of a Reuse-Driven, Domain-Specific System Requirements Analysis Process*. Technischer Bericht Nr. 13/99, Sonderforschungsbereich 501, Fachbereich Informatik, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

[59] Marvin V. Zelkowitz: *A Functional Correctness Model of Program Verification*. IEEE Computer, Seiten 30-39, November 1990.

Teilprojekt B2

Flexible Planung und Ausführung von Softwareentwicklungsprozessen

Leiter

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

Mitarbeiter

Ergänzungsausstattung:

Dipl. Inform. H. Holz
Dipl. Inform. B. Dellen (bis 31.12.98)
Dipl. Inform. M. Schaaf(ab 15.2.99)

Grundausstattung:

Dipl. Inform. F. Weberskirch
Dipl. Inform. S. Schmitt

1 Kurzfassung der Aufgabenstellung

Langfristiges Ziel des Teilprojekts B2 ist die Entwicklung von Methoden und Techniken zur flexiblen Planung und Steuerung komplexer Software-Entwicklungsprozesse. Hierbei liegt das Hauptaugenmerk darauf, Aktivitäten, die sich mit dem Management des Softwareentwicklungsprozesses beschäftigen (also Meta-Aktivitäten), möglichst fließend in die Projektdurchführung zu integrieren. Dazu wurden in der ersten Förderungsperiode Grundlagen entwickelt, welche die Modellierung und Operationalisierung von Projektplänen ermöglichen. Darauf aufbauend sollten in der zweiten Förderungsperiode die folgenden Ziele erreicht werden:

- Weiterentwicklung von Methoden und Techniken zur Verwaltung von Änderungen von Projektplänen während der Ausführung
- Entwicklung von Konzepten zur Herleitung und Verwaltung von domänenspezifischen Abhängigkeiten
- Bereitstellung von Techniken zur Verwaltung von Entscheidungsbegründungen

Hierzu wurde in Zusammenarbeit mit dem Teilprojekt A2 das Prozessunterstützungssystem MILOS[1] implementiert. Es 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.

2 Entwicklung des Teilprojekts in der Förderperiode

Die inhaltlichen Ergebnisse des Teilprojekts werden im folgende Abschnitt erläutert. Hier werden äußere Ereignisse, an denen das Teilprojekt beteiligt war, beschrieben.

Sowohl in 1998 als auch in 1999 wurde auf der Konferenz “Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises” (WETICE) ein Workshop “Coordinating Distributed Software Development Projects” organisiert. In Zusammenarbeit mit Prof. Frank Maurer (University of Calgary, assoziiertes Mitglied des SFB und vormaliger Mitleiter des B2-Projekts) wurde auf der “International Conference on Software Engineering” (ICSE) in 1999 und 2000 je ein Workshop zum Thema “Software Engineering over the Internet” veranstaltet.
In Kooperation mit der Arbeitsgruppe von Prof. Maurer wurde auch der MILOS-Prototyp entwickelt und implementiert. Das System ist 1999 sowohl auf der ICSE als auch auf der SEKE (“International Conference on Software Engineering and Knowledge Engineering”) demonstriert [33] worden.
Die Konzepte der Abhängigkeitsverwaltung in MILOS wurden in enger Zusammenarbeit mit Dr. Charles Petrie (Center for Design Research, Stanford University) sowie

1. Modeling Language and Operational Support for Software Processes

Jussi Stader und Peter Jarvis (Artificial Intelligence Application Institute, University of Edinburgh) diskutiert und weiterentwickelt. Aus der Zusammenarbeit mit Dr. Charles Petrie ging eine gemeinsam betreute Diplomarbeit hervor [15].

3 Durchgeführte Arbeiten und Ergebnisse

Ein zentraler Gegenstand des B2-Projektes war das MILOS-System [40], dessen Konzeption auf den Systemen MVP-L (AG Rombach) und CoMo-Kit (AG Richter) beruht. Die Implementierung geschah dabei in enger Zusammenarbeit mit dem Projekt A2. Die Schwerpunkte von B2 in der MILOS-Entwicklung lassen sich stichwortartig an den erzielten Ergebnissen ablesen:

- Entwicklung und Implementierung einer flexiblen Planungskomponente, die insbesondere auch den Import von Microsoft-Project-Plänen in MILOS erlaubt. Änderungen in einem MS-Project-Plan bewirken eine sofortige Aktualisierung des Zustands der MILOS-Workflowengine beim erneuten Import.
- Entwicklung und Implementierung der Event-Condition-Action- (ECA-) Regeln [20] und Erstellung eines Editors zur Erfassung und Löschung von Regeln als Ergänzung zu den automatisch vom System generierten Abhängigkeiten.
- Darauf basierend die Konzeption eines Notifikationsmechanismus für das Management von Planänderungen.
- Erstellung eines Produktmodells zur Verwaltung von domänenspezifischen Abhängigkeiten innerhalb von Produkten des Softwareentwicklungsprozesses; dies wurde am Beispiel von UML-Klassendiagrammen implementiert [12].

Der Änderungsdienst ist eine wesentliche Komponente eines industriellen Anwendungsprojekts bei der Firma Nortel in Calgary (durchgeführt von Prof. Maurer unter Mitwirkung von Gaststudenten aus Kaiserslautern).
Insgesamt wurden damit wichtige Aspekte der Verschmelzung von Planung und Ausführung realisiert. Ein wesentlicher Gesichtspunkt ist nun, dass viele Änderungsfragen die Planung selbst als Gegenstand haben. Die in dieser Periode vorgenommenen Realisierungen haben dies in die Planungsebene selbst integriert ohne eine explizite Meta-Planung zu erfordern. Für diese Vorgehensweise scheint aber jetzt im wesentlichen eine Grenze erreicht zu sein. Deshalb wird auf den Fortsetzungsantrag hingewiesen, in dem Meta-Planung zum expliziten Gegenstand der Betrachtung gemacht wird (vgl. auch [39]).

3.1 Methoden und Techniken zur Verwaltung von Planänderungen während der Projektdurchführung

Planänderungen während der Projektdurchführung wirken sich direkt auf die Ausführung aus und müssen sich unmittelbar in Aktionen umsetzen lassen. Solche Aktionen sind zunächst Notifizierungen der beteiligten Stellen, die durch ein System von ECA-

Regeln erfolgen. Dabei war die inhaltliche Schwierigkeit, die betroffenen Adressaten mit vertretbarem Aufwand möglichst genau zu ermitteln. Hier galt es, einen Kompromiss zu schließen, der hier so getroffen wurde, dass die Gruppe der von einer Änderung eventuell betroffenen (und daher zu notifizierenden) Benutzer restriktiv aus den dem System bekannten Abhängigkeiten extrahiert wurde. Die zweite wichtige Art von Aktionen betrifft die Änderungen in der Workflowengine, die ebenfalls unmittelbar zu erfolgen haben. Dabei dürfen bisherige Ergebnisse nicht verloren gehen, sie werden in der Object-Version-Management-Komponente von MILOS verwaltet und in dem OODBMS GemStone abgelegt, so dass sie jederzeit wieder zugreifbar sind.

Abbildung 1 demonstriert die Zusammenarbeit zwischen Planer und Planausführern, die vom MILOS-System integriert verwaltet wird.

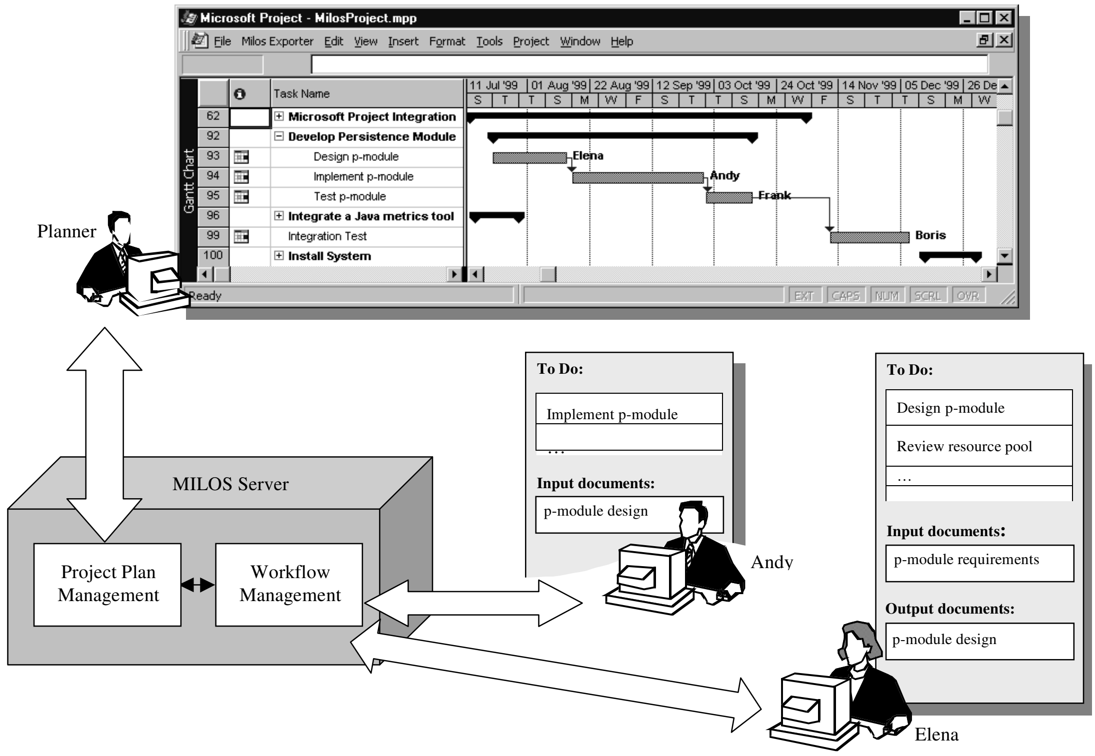

Abbildung 1: Integrierte Verwaltung von Planung und Ausführung in MILOS

3.2 Konzepte zur Herleitung und Verwaltung von domänenspezifischen Abhängigkeiten

Domänenspezifische Abhängigkeiten können auf sehr unterschiedlichen Ebenen formuliert werden. Hier wurde exemplarisch die Ebene der Klassen resp. Klassendiagramme gewählt. Die Abhängigkeiten stellen Wissen dar und betreffen Zusammenhänge, die vom Fachmann erkannt werden müssen und sich nicht aus rein formalen Bezügen (z.B. der Arbeitszuordnung von Projektbeteiligten) ergeben. Solche Abhängigkeiten sind auch in verallgemeinerter Form als Muster darstellbar. Die

Akquisition der Abhängigkeiten erfolgte mit Hilfe der Mitarbeiter der Projekte D1 und D2 und der Teams [10]. Zwei wesentliche Schwierigkeiten bestanden einerseits in der Herausarbeitung der grundsätzlichen Natur dieses Wissens und andererseits dessen Formulierung in der Form, dass ein unmittelbarer Zugriff möglich ist. Der hier erarbeitete Ansatz kann keinen Anspruch auf allgemeine Gültigkeit erheben, weil es sich um ein prinzipielles Problem des Wissensmanagements handelt.
Die identifizierten Abhängigkeiten und Abhängigkeitsmuster wurden in MILOS dargestellt. Zu diesem Zwecke wurde eine XML-Repräsentation entwickelt, welche komplexe Produkte strukturiert und Abhängigkeitsbeziehungen innerhalb von Produkten formalisiert [12].

3.3 Techniken zur Verwaltung von Entscheidungsbegründungen

Ein weiterer Forschungsschwerpunkt dieser Förderungsperiode war, Gründe für getroffene Entscheidungen (*Design Rationales*) dokumentieren und verwalten zu können. Damit werden zwei Ziele verfolgt:

1. Die kausalen Zusammenhänge zwischen getroffenen Entscheidungen einerseits sowie vorausgegangenen Entscheidungen und bereits erzeugten Produkten andererseits sollen protokolliert werden, um im Falle einer Entscheidungsrevision die betroffenen Produkte, Prozesse und Entscheidungen identifizieren zu können.
2. Begründungen für Entscheidungen und die Historie der Entscheidungsfindung sollen dokumentiert werden, um im Nachhinein Gründe für getroffene oder revidierte Entscheidungen rekonstruieren zu können.

In dieser Förderungsperiode wurde das Hauptaugenmerk unserer Forschung auf den ersten dieser Punkte gelegt. Hier war das übergeordnete Ziel, ein effizientes Änderungsmanagement zur Verfügung zu stellen, wie in 3.1 beschrieben. Es stellte sich heraus, dass die explizite Ausformulierung von Entscheidungsbegründungen dazu nur eingeschränkt notwendig war. Um von einer Änderung betroffene Personen möglichst gezielt informieren zu können genügte es vielmehr, die Abhängigkeiten zwischen einzelnen Prozessen (und ihren Lösungsalternativen), Produkten und Entscheidungen in Form von Event-Condition-Action-Regeln darzustellen: Viele Gründe für Entwurfsentscheidungen im Software Engineering betreffen vorliegende Alternativen und ihre Bewertungen, ausgewählte Ereignisse, vorliegende Bedingungen bzw. Constraints, und ihre Beziehungen untereinander.
Für unsere Zwecke konnten Design Rationales also inhaltlich auf die in 3.1 beschriebenen ECA-Regeln abgebildet werden. Eine nachträgliche Extraktion von Entscheidungsbegründungen ist nach wir vor für Dokumentationszwecke von Interesse. Sie wurde jedoch zurückgestellt, da die Dokumentation selbst nicht Gegenstand dieser Förderungsperiode war. Eine derartige Extraktion aus den vorliegenden ECA-Regeln ist jedoch leicht möglich und soll in der nächsten Förderungsperiode im Rahmen der Behandlung von Planungs- und Ausführungstraces wieder aufgegriffen werden. Hier können auch Abfolgen von Entscheidungen (*Entscheidungs-Historien*) in natürlicher Art und Weise angebunden werden.

4 Zusammenarbeit mit anderen Teilprojekten

Wie bereits erwähnt, erfolgte die Implementierung der im Teilprojekt B2 entwickelten Konzepte in MILOS in enger Zusammenarbeit mit dem Teilprojekt A2, welches auch deren Evaluierung vornahm. Dabei wurden Ergebnisse des Teilprojektes B1 zu Konzepten der Planung und Planinstrumentierung berücksichtigt.
Zur Identifikation von domänenspezifischen Abhängigkeiten wurden Ergebnisse der Teilprojekte D1 und D2 und der Teams herangezogen.
Die Ergebnisse des Teilprojekts B2 wurden in Zusammenarbeit mit dem Teilprojekt A1 in die Erfahrungsdatenbank des SFB (SFB-EB) eingebracht.

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] J. Bonnermann: *Ressourcenmanagement in MILOS; Projektarbeit Fachbereich Informatik;* Universität Kaiserslautern, 2000

[2] S. Christofidelli: *Zustandsmodelle in flexiblen Entwicklungsprozessen;* Projektarbeit Fachbereich Informatik, Universität Kaiserslautern, 1998

[3] J. Hohwiller: *Building a virtual marketplace for software development tasks;* Projektarbeit Fachbereich Informatik, Universität Kaiserslautern, 2000

[4] J. Petry: *Vor- und Nachbedingungen für Prozesse in MILOS;* Projektarbeit Fachbereich Informatik, Universität Kaiserslautern, 1999

[5] H. Pharo: *Modellierung von Meta-Aufgaben in MILOS*; Projektarbeit Fachbereich Informatik, Universität Kaiserslautern, 2000

[6] A. Raquet: *Spezifiaktion und Integration von Vor- und Nachbedingungen für Aufgaben in CoMo-Kit*; Projektarbeit Fachbereich Informatik, Universität Kaiserslautern, 1997

[7] B. Schadt: *Realisierung eines Frameworks zur graphischen Darstellung und Editierung von MILOS-Objekten in Java*; Projektarbeit Fachbereich Informatik, Universität Kaiserslautern, 2000

Diplomarbeiten

[8] C. Arenz: *Modellierung von Softwareengineering-Prozessen mit MILOS;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[9] U. Balg: *Wiederverwendung von Entwurfsmustern durch ein Workflow-Management-System;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[10] S. Christofidelli: *Verwaltung von Abhängigkeiten und Entscheidungsbegründungen bei der Planung und Ausführung von Software-Entwicklungsprozessen*; Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, September 2000

[11] A. Herzlinger: *A Distributed Workflow Management Approach for Process-oriented Knowledge Management Systems;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[12] G. Landeck: *XML-basierte Repräsentation von Abhängigkeiten in komplexen Produkten des Software-Entwicklungsprozesses*; Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, Oktober 2000

[13] S. Mungenas: *Entwicklung eines Ansatzes zur verteilten Abarbeitung von Workflows in einem virtuellen Unternehmen*; Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[14] F. Neblung: *Validierung und Erweiterung der Prozessmodellierungssprache MILOS;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[15] A. Raquet: *Dynamic Project Management for Distributed Construction Processes;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[16] T. Rüger: *Behandlung von Risiken in CoMo-Kit;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

[17] M. Schaaf: *Ein System zum Kommunikationsmanagement verteilter Objekte in Java;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[18] M. Schlindwein: *A Framework Using Computer Agents to Provide Automatic Execution Support for Processes in Software Development Projects;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[19] S. Schmitt: *WWW-Based Software Process Modeling with CoMo-Kit;* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1998

Dissertationen

[20] Dellen, Barbara: *Change Impact Analysis Support for Software Development Processes;* Dissertation, Universität Kaiserslautern, Kaiserslautern, Feb. 2000.

Vorträge

[21] B. Dellen: *Internetbasiertes Wissensmanagement in Softwareprojekten*; Wissensmanagement 1999: Strategien der Gegenwart, Industriekonferenz, Frankfurt, 1998

[22] S. Goldmann: *Internetbasierte Prozessmodellierung und Projektplanung*; Wissensmanagement 1999: Strategien der Gegenwart, Industriekonferenz, Frankfurt, 1998

[23] M. M. Richter: *Flexibles Workflow-Management in Software-Entwicklungsprozessen;* Forschungsforum Wissensmanagement und schnelle Produktentwicklung, Universität Stuttgart, 1999

Tagungsbeiträge

[24] F. Bendeck, S. Goldmann, H. Holz, B. Kötting: *Coordinating Management Activities in Distributed Software Development Projects;* Proceedings of the 7 th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 33-38, 1998

[25] B. Dellen, F. Maurer: *Internet Based Software Process Support;* Proceedings of the 7 th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 27-32, 1998

[26] B. Dellen, F. Maurer: *Change Impact Analysis Support for Software Development Processes;* Journal of Applied Software Technology, International Academic Publishing, 1998

[27] B. Dellen, F. Maurer: *A Concept for an Internet-based Process-Oriented Knowledge Management Environment;* Proceedings of the KAW'98, Banff, Canada, 1998

[28] S. Goldmann, J. Münch, H. Holz: *Distributed Process Planning Support with MILOS;* Proceedings of the 11th International Software & Engineering and Knowledge Engineering Conference (SEKE ‘99), Knowledge Systems Institute, ISBN 1-891706-01-2, 1999

[29] S. Goldmann, J. Münch, H. Holz: *A Meta-Model for Distributed Software Development;* Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’99), IEEE Computer Society Press, 1999

[30] S. Goldmann, , J. Münch, H. Holz: MILOS: *A Model of Interleaved Planning, Scheduling, and Enactment;* presented at the ICSE ’99 Workshop on „Software Engineering over the Internet“; Web proceedings only: http://sern.cpsc.ucalgary.ca/~maurer/ICSE99WS/Program.htm

[31] H. Holz, S. Goldmann, F. Maurer: *Working Group Report on Coordinating Distributed Software Development Projects;* Proceedings of the 7 th IEEE International Workshops on Enabling Technologies: Infrastructure for Collab-

orative Enterprises (WET ICE ’98), IEEE Computer Society Press, ISBN 0-8186-8751-7, pp. 69-72, 1998

[32] F. Maurer, H. Holz: *Process-Oriented Knowledge Management For Learning Software Organizations;* Proceedings of the 12th Knowledge Acquisition Workshop (KAW '99), Banff, Canada, 1999

[33] F. Maurer, G. Succi, H. Holz, B. Kötting, S. Goldmann, B. Dellen: *Software Process Support over the Internet;* Proceedings of the 21 th International Conference on Software Engineering (ICSE ’99), ACM Press, ISBN 1-58113-074-0, pp. 642-645, 1999

[34] F. Maurer, H. Holz: *Process-Centered Knowledge Organization for Software Engineering;* Aha, D.W.,& Munoz Avila, H. (Eds.): Exploring Synergies of Knowledge Management and Case-Based Reasoning: A 1999 AAAI Workshop (Working Notes). (Technical Report AIC-99-008). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, 1999

[35] 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

[36] Ch. Petrie, S. Goldmann, A. Raquet: *Agent-Based Project Management;* Proc. of the Intnl.Workshop on Intelligent Agents in CSCW, Deutsche Telekom, Dortmund, Sept. 1998, p. 1-1. (short version of paper of the same name in Springer LNAI 1600)

[37] M.M. Richter, B. Kötting, S. Goldmann: *Flexibles Workflow Management in Software Development Processes;* to appear in Proceedings of the “Forschungsforum Wissensmanagement und schnelle Produktentwicklung”, Stuttgart, Germany, 1999

Zeitschriften / Monografien

[38] S. Goldmann, B. Kötting: *Software Engineering over the Internet;* IEEE Internet Computing, Vol. 3, No 4, July/August 1999, pp 93 – 94, IEEE Computer Society Press, 1999

[39] S. Goldmann, J. Münch, H. Holz: *Distributed Process Planning Support with MILOS;* Int. Journal of Software Engineering and Knowledge Engineering, October 2000 (to appear)

[40] F. Maurer, B. Dellen, F. Bendeck, S. Goldmann, H. Holz, B. Kötting, M. Schaaf: *Merging project planning and web-enabled dynamic workflow for software development;* IEEE Internet Computing, Special Issue on Internet-Based Workflow, May/June 2000 (to appear)

[41] Ch. Petrie, S. Goldmann, A. Raquet: *Agent-Based Project Management;* Springer Lecture Notes on Artificial Intelligence (LNAI) 1600 special volume on “Artificial Intelligence Today”, Springer Verlag, 1998

[42] M. M. Richter: *Knowledge-Based Systems;* Intelligence and Artificial Intelligence (ed. U. Ratsch, M.M. Richter, I.-O. Stamatescu), Springer Verlag, 1998

6 Fremdveröffentlichungen

[43] 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

[44] S. Goldmann, B. Kötting: *Working Group Report of the 2nd Workshop on Coordinating Distributed Software Development Projects;* Proceedings of the 8th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’99), IEEE Computer Society Press, 1999

[45] John Grundy, John Hosking, Rick Mugridge: *Coordinating Distributed Software Development Projects with Integrated Process Modelling and Enactment Environments;* in IEEE Proceedings of the 7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, USA, 1998

[46] P. Jarvis, J. Stader, A. Macintosh, J. Moore, P. Chung: *What Right Do You Have to Do That? Infusing Adaptive Workflow Technology with Knowledge about the Organisational and Authority Context of a Task;* Proceedings of First International Conference on Enterprise Information Systems (ICEIS-99), Setubal, Portugal, March 1999

[47] B. Kötting, F. Maurer: *Approaching Process Support for Virtual Software Corporations;* presented at the ICSE ’99 Workshop on Software Engineering over the Internet; Web proceedings only: http://sern.cpsc.ucalgary.ca/~maurer/ICSE99WS/Program.htm

[48] B. Kötting, F. Maurer: *A Concept for Supporting the Formation of Virtual Corporations through Negotiation;* Proceedings of the 8 th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE ’99), IEEE Computer Society Press, 1999

Teilprojekt B4

Generische Kommunikationssysteme

Leiter

Prof. Dr. Reinhard Gotzhein
geb. 9.10.1956
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-3426
Fax: 0631/205-3956
gotzhein@informatik.uni-kl.de

Mitarbeiter

Ergänzungsausstattung:

Dipl. Inform. B. Geppert (bis 15.5.1999)
Dipl. Inform. C. Peper
Dipl. Inform. J. Thees (seit 1.2.2000)

Grundausstattung:

Dipl. Inform. F. Rößler (bis 15.5.1999)
Dipl. Inform. P. Schaible (seit 15.5.1999)

1 Kurzfassung der Aufgabenstellung

Langfristiges Ziel dieses Teilprojekts ist die Bereitstellung 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 der Ergebnisse wird durch die Bereitstellung und den Einsatz domänenspezifischer Methoden, formaler Techniken und rechnergestützter Werkzeuge sichergestellt [53].

Zur Erreichung der langfristigen Ziele des Teilprojekts wurden in der Förderperiode die folgenden Forschungsschwerpunkte bearbeitet:

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”.

2 Entwicklung des Teilprojekts in der Förderperiode

Exponate auf Industriemessen

Einige der wissenschaftlichen Ergebnisse des Teilprojekts wurden in Exponaten auf der CeBIT’98 [58] [59] sowie auf dem Technologietransferkongress “IKT-Anwendungen auf Netzen” im Rahmen des III. Mannheimer Unternehmerforums [60] vorgestellt.

Präsentationen auf internationalen Veranstaltungen

An exponierter Stelle konnte der im Rahmen des Teilprojekts entwickelte SDL-Pattern-Ansatz auf zwei internationalen Veranstaltungen der SDL-Community, die traditionell eine besonders starke Industriebeteiligung aufweisen, präsentiert werden:

- Während der Internationalen Konferenz “*9th SDL Forum 1999*” wurde das Tutorial “*Pattern-Based Reuse with SDL*” gehalten [37].
- Im Rahmen des Internationalen Workshops “*SDL and MSC (SAM’98)*” wurde die halbtägige Sitzung “*Patterns and SDL*” organisiert und geleitet (eingeladener Sprecher: Erich Gamma) [62].

Guest Editor “Journal of Universal Computer Science”

In diesem Jahr erscheint eine Sonderausgabe der Internationalen Zeitschrift *Journal of Universal Computer Science* (kurz: *J.UCS*), die einen besonderen thematischen

Bezug zum SFB 501 besitzt [54]. Das Thema der Sonderausgabe lautet "*Requirements Engineering: The Light Control Case Study*", Herausgeber sind Egon Börger und Reinhard Gotzhein. Der Call for Papers (URL: http://rn.informatik.uni-kl.de/~recs) fand im September 1999 statt, zum Berichtszeitpunkt sind die Begutachtungsphase und die Auswahl der akzeptierten Beiträge abgeschlossen. Der besondere SFB-Bezug besteht in der Wahl der Fallstudie, die auf einer Fallstudie aus den Querschnittprojekten Q1 und Q4 basiert. Damit werden die im Requirements Engineering erzielten Ergebnisse des SFB international vergleichbar.

Associate Rapporteur der ITU-T "Formal Semantics Definition for SDL-2000"

Während des ITU-T SG10 Meetings in Genf wurde eine revidierte Fassung der Norm Z.100, welche die im SFB in mehreren Teil- und Querschnittprojekten eingesetzte Entwurfssprache SDL definiert, verabschiedet. Die Arbeit an der formalen Semantik von SDL-2000 begann Anfang 1998 [43] und wird seither vom Projektleiter des Teilprojekts B4 in der Funktion des Associate Rapporteurs mitgetragen. Im selben Jahr fiel die Entscheidung, den Formalismus der Abstract State Machines (ASMs) zur operationalen Definition der formalen Semantik einzusetzen [42] [14]. Die Fertigstellung der Semantik ist für November 2000 vorgesehen.

3 Durchgeführte Arbeiten und Ergebnisse

3.1 Arbeitsschwerpunkt A: Generische Kommunikationsdienste

Arbeitspaket A1: Methoden und Techniken

Prozessmodell. Das in der 1. Förderperiode entwickelte erste Prozessmodell für die Problemanalyse wurde aufgrund der Erfahrungen mehrerer Fallstudien (s. Arbeitspaket A3) weiterentwickelt und konsolidiert [44] [48]. An dieser Stelle seien vier Aspekte des Prozessmodells besonders hervorgehoben:

- Die Problemanalysephase verläuft iterativ, wobei jede Iteration neben der Formalisierung ein Review mit Kundenbeteiligung auf der Basis einer schematisch übersetzten, semiformalen Fassung vorsieht.
- Ergebnis der Analysephase ist eine formale Problemspezifikation, die direkt als Eingabe für die Entwurfsphase dient. Diese Spezifikation liegt bei positivem Abschluss des Reviews mit Kundenbeteiligung bereits fertig vor.
- Zur Formalisierung der Problembeschreibung werden musterbasierte Ansätze verwendet.
- Das Prozessmodell beinhaltet ferner Schritte zur Gewinnung von Mustern durch Analyse von Entwicklungsdokumenten abgeschlossener Projekte [44].

Das Prozessmodell ist unabhängig von der eingesetzten Spezifikationstechnik. Es wurde zur Durchführung von Fallstudien mit der in Kooperation mit Teilprojekt C1 entwickelten Spezifikationstechnik FoReST [57] instanziiert.

Spezifikationstechniken. In der ersten Förderperiode wurde zur formalen Spezifikation von Anwendungsanforderungen und Kommunikationsdiensten die auf das Anwendungsfeld zugeschnittene Realzeit-Logik tRTTL (tailored Real Time Temporal Logic) eingesetzt. Die beim Einsatz von tRTTL auftretenden Defizite hinsichtlich Strukturierung und Ermittlung von Kommunikationsanforderungen konnten mit der Entwicklung der *Formal Requirements Specification Technique* (*FoReST)* [57] [56] [16] (in Kooperation mit Teilprojekt C1), die objektorientierte Konzepte und tRTTL integriert, beseitigt werden.

Referenzmodell für Problemspezifikationen. Aufbauend auf dem in [65] vorgestellten Referenzmodell für Anforderungen und Spezifikationen wurde in Kooperation mit Teilprojekt C1 ein Referenzmodell für Problemspezifikationen definiert [46] [57]. Das Referenzmodell unterscheidet zwischen beobachteten und kontrollierten Größen sowie zwischen Umgebungs- und Systemgrößen und trifft darüberhinaus Festlegungen bezüglich ihrer Sichtbarkeit.

Ermittlung der Kommunikationsanforderungen. Zur Ermittlung der Kommunikationsanforderungen wird die Problemspezifikation - unterstützt durch den Einsatz von *Service Patterns* - verfeinert. Über die Definition der Service Patterns, insbesondere das Analysemodell als Bestandteil dieser Definition, kann ein unmittelbarer Zusammenhang zu Requirement Patterns hergestellt werden. Damit ist aus dem Einsatz eines Requirement Patterns in der Problemspezifikation ein direkter Schluss auf die zur Verfeinerung anwendbaren Service Patterns möglich. Die Adaption des Service Patterns ist teilweise aus der bereits gegebenen Adaption des Requirement Patterns ermittelbar. Die Verfeinerung kann in mehreren Schritten erfolgen [47]. Die praktische Erprobung dieser Vorgehensweise sowie der Aufbau einer initialen Service-Pattern-Sammlung sind im Gange und sollen noch in dieser Förderperiode zu einem abschließenden Ergebnis führen.

Arbeitspaket A2: Verbreiterung und Verwaltung der Dienstmustersammlung

Requirement-Pattern-Sammlung. Im Berichtszeitraum wurden mehrere umfangreiche Fallstudien zur Problemspezifikation durchgeführt (s. Arbeitspaket A3), bei denen die Requirement-Pattern-Sammlung intensiv eingesetzt wurde. Im Anschluss an diese Fallstudien fand jeweils eine Überarbeitung der Mustersammlung statt. So hat sich die Sammlung gegenüber der initialen Version stark verändert, auch wenn sich die Anzahl der Muster kaum erhöht hat. Verschiedene Stadien von Requirement Patterns sind in [44], [48] und [15] dokumentiert. In [44] wird die Gewinnung von Requirement Patterns im allgemeinen und des LazyReaction-Patterns im besonderen ausführlich dargestellt.

Service-Pattern-Sammlung. Die Entwicklung der Service-Pattern-Sammlung ist wie bereits erörtert eng an die Requirement Patterns gekoppelt. Aufgrund dieser starken Abhängigkeiten und bedingt durch die neuen Resultate im Anforderungsbereich ist der Aufbau der Service-Pattern-Sammlung noch im Gange, so dass hier erst gegen Ende der laufenden Förderperiode aussagekräftige Ergebnisse vorliegen werden.

Werkzeugunterstützung. Die aktuelle Requirement-Pattern-Sammlung ist im organisatorischen Teil der Erfahrungsdatenbank SFB-EDB (s. Teilprojekt A1) abgelegt und kann bei der Durchführung von Projekten herangezogen werden. Ferner wurde das in Teilprojekt C1 entwickelte rechnergestützte Werkzeug *xforest* [64] zur Verwaltung der Requirement Patterns eingesetzt.

Arbeitspaket A3: Engineering von maßgeschneiderten Kommunikationsdiensten im SFB-Anwendungsfeld

Die in den Arbeitspaketen A1 und A2 entwickelten Methoden, Techniken, Werkzeuge und Mustersammlungen wurden in mehreren umfangreichen Fallstudien [57] [45] [18] (s. Querschnittprojekt Q4) sowie einer Variante der Problemspezifikation "Lichtsteuerung", die als Ausgangspunkt für die durchgängige Entwicklung einer verteilten Gebäudesteuerung dient, eingesetzt und erprobt. Die darin gesammelten Erfahrungen sind jeweils unmittelbar in die Weiterentwicklung und Konsolidierung dieser Ansätze eingeflossen.

3.2 Arbeitsschwerpunkt B: Generische Kommunikationsprotokolle

Arbeitspaket B1: Modellierung und Erfassung der Kompositionsintelligenz

Für die Klasse der Interaktionsmuster wurden ein Kompositionsmodell sowie Sprach- und Werkzeugunterstützung entwickelt.

Kompositionsmodell. Der Protokollentwurf mit SDL-Patterns unterscheidet sich von traditionellen SDL-Entwurfsmethoden durch eine kollaborationsorientierte Systemsicht [37]. Kollaborationen sind im Sinne von Systemfunktionalitäten (z.B. Verbindungsaufbau oder Datentransfer) zu verstehen und stellen damit nur einen Teil des Gesamtsystems dar. Sie sind ein wesentliches Strukturierungsmittel zur Beschreibung verteilter Systeme und müssen vom Protokollingenieur entsprechend der funktionalen und nicht-funktionalen Systemanforderungen entworfen werden.

Sprachunterstützung. Zur Verbesserung des Entwurfsprozesses wurde die Notation CoSDL entwickelt, die es erlaubt, initiale Entwurfsideen in Form von Kollaborationsdiagrammen zu beschreiben [37]. CoSDL ist eine graphische Notation, die sich stark an SDL anlehnt und (ähnlich zu MSC) Interaktionsverhalten zwischen SDL-Prozessen beschreibt. Im Gegensatz zu MSC basiert CoSDL aber nicht auf der Spezifikation von Beispielabläufen, sondern beschreibt vollständige Kollaborationen zwischen SDL-Prozessen. Im Prozessmodell des SDL-Pattern-Ansatzes werden Kollaborationsdiagramme während der Systemanalyse eingesetzt und unterstützen die Selektion von SDL-Patterns. CoSDL wurde beim Re-Engineering eines ATM-Signalisierungsprotokolls erprobt [4].

Werkzeugunterstützung. Das prototypische Werkzeug SPEEDI (s. Arbeitspaket B2) unterstützt die kollaborationsorientierte Systemsicht bereits teilweise.

Arbeitspaket B2: Verbreiterung und Bewertung des Protokollbaukastensystems

Das Protokollbaukastensystem besteht aus einem inkrementellen, musterbasierten Entwurfsprozess sowie einer SDL-Pattern-Sammlung. Im Berichtszeitraum sollte das initiale Protokollbaukastensystem [35] [36] unter anderem durch eine geeignete Werkzeugunterstützung ausgebaut und darüber hinaus bewertet werden.

Musterbasierter Entwurfsprozess. Ausgangspunkt für den Protokollentwurf sind Kommunikationsanforderungen, die in einem ersten Schritt mit dem Ziel der iterativen Behandlung in Teilanforderungen dekomponiert werden. Mit jedem nun folgenden Durchlauf werden weitere Teilanforderungen analysiert und in einen SDL-Entwurf überführt. Aufgabe der Kollaborationsanalyse ist dabei die Erstellung einer konzeptionellen Systemarchitektur (bzw. die Ergänzung der Architektur des letzten Entwicklungsschrittes) sowie die Identifikation und Modellierung von Kollaborationen zwischen den vorhandenen Architekturelementen. Der eigentliche Entwurfsschritt wird ebenfalls inkrementell durchgeführt, wobei die zuvor identifizierten Kollaborationen schrittweise in den vorhandenen SDL-Entwurf integriert werden. Dies kann durch Anwendung von SDL-Patterns oder konventionell, ohne Musterunterstützung, geschehen. Die SDL-Pattern-Anwendung besteht ihrerseits aus den Teilschritten Selektion, Adaption und Komposition. Das Resultat eines Durchlaufs ist eine ausführbare SDL-Entwurfsspezifikation, die validiert und in nachfolgenden Durchläufen ergänzt wird. Sind alle Teilanforderungen abgearbeitet, ist der inkrementelle Entwurfsprozess abgeschlossen, und die resultierende validierte SDL-Entwurfsspezifikation dient als Eingabe für die nachfolgende Implementierungsphase.

SDL-Pattern-Sammlung und Beschreibungsrahmen. Der in der ersten Förderperiode definierte Beschreibungsrahmen für SDL-Patterns [33] wurde durch die Einführung von Metasymbolen zur Beschreibung der SDL-Fragmente weiterentwickelt. Die bisherige Notation unterstützte lediglich die Darstellung der invarianten Teile einer Entwurfslösung adäquat, nicht jedoch die Darstellung der zulässigen Adaptionen an einen konkreten Kontext. Die eingeführten Metasymbole erfassen die erlaubten Variationen und werden bei der Adaption eines Musters aufgelöst, so dass ein komponierbares SDL-Fragment entsteht. Die SDL-Pattern-Sammlung selbst wurde aufgrund der in mehreren Fallstudien gesammelten Erfahrungen weiterentwickelt und ausgebaut (s. Arbeitspaket B3). Ferner wurde der Zusammenhang zwischen der in der objektorientierten Modellierung verwendeten Spezialisierung und der Anwendung von SDL-Patterns untersucht [40].

Evaluierung und Quality Improvement Paradigm. Zur Evaluierung und systematischen Verbesserung wurde der SDL-Pattern-Ansatz gemäß des Quality Improvement Paradigms (QIP) erweitert [31] [32]. In Kooperation mit Teilprojekt A1 wurde ein Messprogramm zur Erfassung des Verbesserungs- und Wiederverwendungspotenzials konzipiert [10] [11] [12]. Die während der Durchführung von Fallstudien mit Hilfe von maßgeschneiderten Fragebögen erfassten Messdaten wurden analysiert, die Ergebnisse dienten als Ausgangspunkt für die Verbesserung des Ansatzes. Unter anderem wurden verschieden Maturity-Levels für SDL-Patterns definiert [38]. Die gesammelten Erfahrungselemente wurden in der SFB-EDB verwaltet. Weiterhin wurden die durchgeführten Fallstudien mit ihren Messdaten in der SFB-EDB dokumentiert. Der SDL-Pattern-Ansatz wird seit 1998 im Praktikum “Entwicklung von Kommunikationssystemen” der AG Rechnernetze eingesetzt und dort in diesem Jahr in Kooperation mit Teilprojekt A1 evaluiert.

Werkzeugunterstützung. Zur Unterstützung der musterbasierten Protokollentwicklung wurde das rechnergestützte Werkzeug *SPEEDI* (SDL-PattErn EDItor) realisiert [29] [56]. SPEEDI unterstützt die *Anwendung* (Selektion, Adaption und Einbettung) von SDL-Patterns in SDL-Spezifikationen auf der Basis von SDL-PR [8]. Ferner *dokumentiert* SPEEDI die Reihenfolge und Platzierung von Musteranwendungen [3], wodurch die orthogonale, musterbasierte Struktur der SDL-Spezifikation festgehalten wird. Das *Browsen* durch den integrierten SDL-Pattern-Pool wird durch SPEEDI bereits rudimentär unterstützt [8]. Weitere Unterstützung wird durch die Einlagerung der Mustersammlung in die SFB-EDB geboten [11].

Arbeitspaket B3: Engineering von maßgeschneiderten Protokollen im SFB-Anwendungsfeld

Gegenstand dieses Arbeitspaketes ist die Durchführung von Fallstudien zur Evaluierung und Verbesserung des SDL-Pattern-Ansatzes. Zur Bewertung wurde zunächst eine *informelle* Evaluierung anhand folgender Fallstudien durchgeführt:

Neuentwicklung von Protokollen. Entwickelt wurden Kommunikationssubsysteme auf der Basis von Token Ring und CAN [7]. Diese Fallstudien dienten als Machbarkeitsstudien für den Einsatz des SDL-Pattern-Ansatzes.

Reengineering von Protokollen. Betrachtet wurden Protokolle aus der Internet Protocol Suite: IPv6 [1], RTP [2] und ST2+ [6] [13]. Diese Fallstudien dienten der Untersuchung, ob bereits bestehende Protokolle mit ihren integrierten Entwurfslösungen durch das Protokollbaukastensystem erfasst sind bzw. ob wesentliche Lücken in der SDL-Pattern-Sammlung bestehen.

Reengineering von SDL-Spezifikationen. Betrachtet wurde das ATM-Signalisierungsprotokoll (UNI 3.1) [4]. In dieser Fallstudie konnte gezeigt werden, dass eine musterbasierte Spezifikation wesentlich zur Systematisierung und Verständlichkeit beiträgt. Es stellte sich außerdem heraus, dass die in der Mustersammlung angebotenen SDL-Entwurfslösungen im wesentlichen denen entsprechen, die von externen SDL-Designern verwendet worden sind.

Die Reihenfolge der durchgeführten Fallstudien ist durch einen zunehmenden Komplexitätsgrad sowie zunehmende Durchgängigkeit des Ansatzes gekennzeichnet. Die gewonnen Erkenntnisse wurden informell festgehalten und flossen jeweils in eine Weiterentwicklung des Ansatzes ein. Beispiele hierfür sind Verbesserungen in der Beschreibung bestehender Muster, Verfeinerung von Prozessschritten oder auch die Entwicklung und Einführung einer Notation zur Beschreibung von Kollaborationen.

Arbeitspaket B4: Automatische Kodegenerierung für maßgeschneiderte Protokolle

Zur prototypischen Implementierung von Kommunikationssubsystemen wurde der Kodegenerator *EnvGen (Environment Generator)* [34] [5] entwickelt. EnvGen ermöglicht in Kombination mit dem kommerziellen Kodegenerator SDT-Cadvanced eine vollautomatische Implementierung und Einbettung von SDL-Spezifikationen. Dazu werden direkt aus der SDL-Spezifikation Umgebungsfunktionen für die Laufzeitbibliothek von SDT-Cadvanced generiert, mit deren Hilfe offene SDL-Systeme über existierende Basistechnologien kommunizieren. Derzeit wird die Anbindung an ATM, CAN, QNX-IPC und BSD-Sockets unterstützt, die Anbindung an UART ist in Vorbereitung. Weitere Basistechnologien können bei Bedarf integriert werden. Ein Teilergebnis dieser Arbeit zeigt, dass sich das im Teilprojekt etablierte Pattern-Konzept auch für den Bereich der Umgebungsschnittstelle einsetzen läßt [56] [52]. In Ergänzung zu den in Arbeitspaket B3 beschriebenen Fallstudien wurden prototypische Implementierungen des objektbasierten Kommunikationssubsystems auf der Basis von CAN [34] und eines ATM-Signalisierungsprotokolls (UNI 3.1) [4] automatisch generiert und getestet.

3.3 Arbeitsschwerpunkt C: Echtzeitfähige generische Kommunikationsprotokolle

Arbeitspaket C1: Integration von QoS-Architekturkomponenten in das Protokollbaukastensystem

Das Ergebnis der Analyse von Gemeinsamkeiten zwischen Echtzeitkommunikationssystemen wurde in der ersten Förderperiode in einem allgemeinen QoS-Architekturmodell für Kommunikationssysteme festgehalten. Danach umfassen QoS-Architekturen im wesentlichen Komponenten für die Benutzerinteraktion, die Reservierung der räumlich verteilten Ressourcen, die Zugangskontrolle, das QoS-Monitoring, die Verkehrskontrolle und das Scheduling. Für jede dieser Architekturkomponenten lässt sich in der Literatur eine Fülle von Mechanismen zur Realisierung nachweisen. Der Versuch, eine Auswahl dieser Mechanismen in das Protokollbaukastensystem zu integrieren, hat unterschiedliche Erkenntnisse erbracht:

- Durch mehrere Fallstudien zur Konfigurierung von Ressourcenreservierungsprotokollen [34] [41] [50] konnte der Protokollbaukasten soweit ausgebaut werden, dass in der aktuellen Ausbaustufe eine ausreichende Überdeckung der SDL-Entwurfsspezifikationen erzielt wird. Eine weitere Spezialisierung der Protokollbausteine für Ressourcenreservierungsprotokolle hätte ein zu geringes Wiederverwendungspotenzial zur Folge.
- Das Protokollbaukastensystem unterstützt die Erstellung von SDL-Entwurfsspezifikationen und ist insofern auf die Entwicklung verteilter, nebenläufiger Systeme ausgerichtet. Einige Komponenten der QoS-Architektur wie die Zugangskontrolle, die Verkehrskontrolle und das Scheduling haben jedoch einen vorwiegend algorithmischen, sequentiellen Charakter und werden daher nicht unter Verwendung von SDL konzipiert. Aus diesem Grund können sie

nicht durch den Protokollbaukasten in seiner jetzigen Konzeption abgedeckt werden.

- Der Protokollbaukasten dient zur generischen Auslegung des Entwicklungsprozesses von (Echtzeit-)Kommunikationssubsystemen, indem Architekturstile und Entwurfslösungen mit hohem Wiederverwendungspotenzial in einem Pool gesammelt und verbessert werden. Das konfigurierte Produkt selbst hat keinen besonders ausgeprägten generischen Charakter. Der Versuch, das QoS-Achitekturmodell (bzw. QoS-Komponenten) in den Protokollbaukasten zu integrieren, hat zu einem verbesserten Ansatz geführt, der auch eine generische Auslegung der konfigurierten Produkte vorsieht. Ziel ist hier die Entwicklung einer generischen Architektur, die es erlaubt, eine Familie vorbestimmter (Echtzeit-)Kommunikationssubsysteme zu instanziieren.

Arbeitspaket C2: Analyse des deterministischen Zeitverhaltens und Effizienzsteigerungsmaßnahmen für die Konfigurierung von zeitkritischen Kommunikationsprotokollen

In Arbeitspaket C2 sollten Verfahren zur Analyse der Vorhersagbarkeit des Zeitverhaltens und Maßnahmen zur Effizienzsteigerung konfigurierter Protokolle entwickelt werden. Aufgrund des Weggangs zweier Mitarbeiter des Teilprojekts B4 im Mai 1999 konnte dieses Arbeitspaket nicht wie vorgesehen bearbeitet werden.

Arbeitspaket C3: Fallstudien zum Entwurf von Echtzeitprotokollen mit generischen Methoden

Um die Konfigurierung von Protokollspezifikationen durch die Implementierung von Prototypen absichern zu können, wurde die Auswahl der aus der ersten Förderperiode zur Verfügung stehenden Kommunikationsbasistechnologien [51] ergänzt. Die neu eingesetzten Basistechnologien (ATM, CAN) wurden wo nötig zu im Sinne des QoS-Architekturmodells vollständigen Echtzeitkommunikationssystemen für die im SFB eingesetzte Zielplattform ausgebaut [34] [49].

Für den CAN-Feldbus wurde ein echtzeitfähiges Kommunikationssubsystem entwikkelt, das im Gegensatz zu herkömmlichen Installationen zur Laufzeit ein dynamisches Ressourcenmanagement erlaubt [7]. Als wesentliche Architekturkomponenten wurden ein objektbasiertes Kommunikationsprotokoll, Netzwerkmanagementprotokolle zur Knoteninitialisierung und Ressourcenreservierung, Quality-of-Service Mechanismen zur Verkehrskontrolle und zum Scheduling sowie ein Gerätetreiber zur Kommunikation mit der CAN-Netzwerkkarte entwickelt. Die Ressourcenmanagementanteile wurden mit Hilfe des Protokollbaukastensystems entwickelt.

4 Zusammenarbeit mit anderen Teilprojekten

In Kooperation mit **Teilprojekt A1** wurde der SDL-Pattern-Ansatz in die SFB-EDB integriert. Auf methodischer Seite ist die Konzipierung und Erstellung eines konkreten Messprogramms auf Basis von GQM zur Evaluierung und Verbesserung des SDL-

Pattern-Ansatzes hervorzuheben [10] [11] [12], das in einer Fallstudie [2] eingesetzt wurde. Die Integration des Messprogramms in den SDL-Pattern-Ansatz resultiert zunächst in einer Erweiterung des SDL-Pattern-basierten Entwurfsprozesses um einen Verbesserungszyklus [31] [32] [38]. Weiterhin wurde die SFB-EDB um die Definition SDL-Pattern-spezifischer Ablagestrukturen angereichert und mit Inhalten (Prozessmodell, SDL-Pattern-Sammlung etc.) gefüllt. Dies wurde zunächst für den Web-basierten Prototypen der SFB-EDB durchgeführt [39]; eine Umstellung der SFB-EDB auf objektrelationale Datenbanktechnologie wird in **Teilprojekt A3** vorgenommen, wobei der SDL-Pattern-Ansatz als Anwendungsbeispiel dient [30] [9]. Der aus der A1/B4-Kooperation resultierende Teil der SFB-EDB wurde in der SDL-Community im Rahmen einer Tooldemonstration [61] sowie eines Tutorials [37] vorgestellt.

Die im SFB verwendeten Entwicklungsmethodiken werden zur Einlagerung in die SFB-EDB von **Teilprojekt B1** in Form formaler Prozessmodelle erfasst. Für SDL-spezifische Entwicklungsprozesse - u.a. der SDL-Pattern-basierte Entwurfsprozess oder der von Telelogic unterstützte SOMT-Prozess - geschah dies unter Nutzung des Expertenwissens von B4 [17].

Mit **Teilprojekt B5** wurde ein SFB-Praktikum konzipiert und durchgeführt, das die Ansätze beider Teilprojekte beim Entwurf und der Implementierung verteilter Systeme integriert. Zum Einsatz kamen dort der SDL-Pattern-Ansatz, kombiniert mit einer anschließenden Kodegenerierung und Anbindungsmechanismen an betriebssystemtechnische Aspekte. Ferner fand während der gesamten Förderperiode eine enge Abstimmung der Teilprojekte mit dem Ziel, die Entwicklung von Kommunikations- und Betriebssystemkomponenten zu verzahnen und zu synchronisieren, statt.

In Kooperation mit **Teilprojekt C1** wurde die Spezifikationstechnik FoReST entwickelt und in mehreren Fallstudien eingesetzt. Diese Aktivitäten sind in Arbeitsschwerpunkt A sowie in Querschnittprojekt Q4 näher beschrieben.

In Abstimmung mit **Teilprojekt D1** wurde im Rahmen einer gemeinsam betreuten Diplomarbeit [7] ein Echtzeitkommunikationssubsystems auf Basis von CAN entwikkelt. Die Aufgabe von B4 bestand in der musterbasierten Entwicklung des Subsystems [41], hierbei insbesondere der objektbasierten Kommunikation auf Anwendungsebene, desweiteren aus der Anbindung an das QoS-Architekturmodell [49], sowie der Kodegenerierung [34]. Das Subsystem wurde zur verteilten Steuerung eines simulierten Inversen Pendels eingesetzt und auf der CeBIT präsentiert [59].

Ein Schwerpunkt der Kooperation bestand durch die Mitarbeit in den **Querschnittprojekten**, in denen alle Teilprojekte Mitarbeiterzeit zur Verfügung gestellt haben. Die Ergebnisse dieser Teams sind in eigenen Berichten dargestellt.

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] Schwaiger, M.: *Musterbasierte Konfigurierung des Internet-Protokolls IPv6 mit SDL*, Dezember 1997

[2] van Lier, S.: *SDL-Pattern basiertes Reengineering von RTP (Realtime Transport Protocol)*, August 1998

Diplomarbeiten

[3] Cisowski, D.: *Ein Editor zur Dokumentation von SDL Patterns*, März 1999

[4] Groß, T.: *Re-Engineering und Implementierung eines ATM-Signalisierungsprotokolls mit SDL-Patterns*, September 1999

[5] Kühlmeyer, A.: *SDT-basierte Anbindung von SDL-Entwurfsspezifikationen an verschiedene Basistechnologien*, März 1999

[6] Schaible, P.: *Pattern-basierte Konfigurierung des Reservierungsprotokolls ST2+ und Erstellung eines SDL-Frameworks für Ressourcenreservierungsprotokolle*, September 1997

[7] Schneider, M.: *Entwurf und Implementierung eines echtzeitfähigen Kommunikationssubsystems für CAN*, Februar 1998

[8] Schwaiger, M.: *Werkzeugunterstützte Anwendung von SDL Patterns*, März 1999

SFB-Berichte

[9] Feldmann, R., Geppert, B., Mahnke, W., Ritter, N., Rößler, F.: *The ORDBMS-Based SFB 501 Experience Base - Exemplified by the SDL-Pattern Approach*, SFB 501 Report 08/99, Fachbereich Informatik, Universität Kaiserslautern, 1999

[10] Feldmann, R., Geppert, B., Rößler, F.: *Towards an Experimental Evaluation of SDL-Pattern based Protocol Design*, SFB 501 Report 04/98, Fachbereich Informatik, Universität Kaiserslautern, 1998

[11] Feldmann, R., Geppert, B., Rößler, F.: *A Repository-Based Improvement Cycle for Reuse-Driven SDL System Development*, SFB 501 Report 11/98, Fachbereich Informatik, Universität Kaiserslautern, 1998

[12] Feldmann, R., Geppert, B., Rößler, F.: *First Results from an Experimental Evaluation of SDL-Pattern Based Protocol Design*, SFB 501 Report 03/99, Fachbereich Informatik, Universität Kaiserslautern, 1999

[13] Geppert, B., Rößler, F., Schaible, P.: *Pattern-based Configuring of the Internet Stream Protocol ST2+ with SDL*, SFB 501 Report 12/97, Fachbereich Informatik, Universität Kaiserslautern, 1997

[14] Glässer, U., Gotzhein, R., Prinz, A.: *SDL Formal Semantics Definition*, Technischer Bericht tr-rsfb-99-065, Fachbereich Mathematik und Informatik, Universität Paderborn, 1999

[15] Gotzhein, R., Kronenburg, M., Peper, C.: *Reuse in Requirements Engineering: Discovery and Application of a Real-Time Requirement Pattern*, SFB 501 Report 8/98, Fachbereich Informatik, Universität Kaiserslautern, 1998

[16] Kronenburg, M., Peper, C.: *An Example of a FoReST Problem Specification*, SFB 501 Report 01/1999, Fachbereich Informatik, Universität Kaiserslautern, 1999

[17] Petersen, E., Münch, J., Geppert, B.: *Ein formales Prozeßmodell für die Software-Entwicklungsmethode SOMT*, SFB 501 Report 03/98, Fachbereich Informatik, Universität Kaiserslautern, 1998

Online-Dokumente

[18] Kronenburg, M., Peper, C.: *Specification of the Case Study “Light Control System”*, Dagstuhl-Seminar Requirements Capture, Documentation, and Validation, 13.-18. Juni 99, http://rn.informatik.uni-kl.de/~forest/examples

Vorträge

[19] Geppert, B.: *SDL Patterns - Ein Ansatz zur generischen Protokollentwicklung,* Forschungsseminar, Humboldt Universität Berlin, 17.2.1998

[20] Geppert, B.: *Der SDL-Pattern-Ansatz*, Workshop Software-Reuse des SFB 501, Kaiserslautern, 22.-23. Februar 1999

[21] Gotzhein, R.: *Formale Spezifikation von Echtzeitanforderungen*, GI-Fachgespräch über Quality of Service, Mannheim, 12.-13.3.1998

[22] Gotzhein, R., Geppert, B., Rößler, F.: *Musterbasierte Entwicklung von Kommunikationsprotokollen mit SDL*, RIO6-Kolloquium, Daimler-Benz Aerospace AG, Bremen, 16.6.1998

[23] Gotzhein, R.: *Engineering-like Development of Customized Communication Software*, Kolloquium, Universität Zürich, 25.6.1998

[24] Gotzhein, R.: *Musterbasierte Systementwicklung*, Daimler-Benz Research, Berlin, 2.7.98

[25] Gotzhein, R.: *Pattern-based System Development*, Kolloquium, Ecole Polytechnique Federale, Lausanne, 9.7.98

[26] Gotzhein, R.: *Wiederverwendung von Lösungskomponenten und Entwicklungs-Know-how*, Deutsche Telekom, Saarbrücken, 16.12.1998

[27] Gotzhein, R.: *Musterbasierte Entwicklung von Kommunikationsmiddleware*, Kolloquium, Fachbereich Informatik, Universität Braunschweig, 12. Juli 1999

[28] Peper, C.: *Wiederverwendung bei der Anforderungsspezifikation: Der FoReST-Ansatz*, Workshop "Software-Reuse" des SFB 501, Kaiserslautern, 22.-23. Februar 1999

Tagungsbeiträge

[29] Cisowski, D. , Geppert, B., Rößler, F., Schwaiger, M.: *Tool Support for SDL Patterns*, Proceedings of the 1st Workshop of the SDL Forum Society on SDL and MSC (SAM98), Berlin, Juni 1998

[30] Feldmann, R., Geppert, B., Mahnke, W., Ritter, N., Rößler, F.: *An ORDBMS-Based Reuse Repository Supporting the Quality Improvement Paradigm - Exemplified by the SDL-Pattern Approach*, in: Proceedings of the TOOLS USA 2000, 34th International Conference & Exhibition, Santa Barbara, USA, July/August 2000

[31] 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

[32] 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

[33] 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

[34] 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 on Formal Description Techniques & Protocol Specification, Testing, and Verification (FORTE/PSTV'98), Paris, Frankreich, Kluwer Academic Publishers, November 1998

[35] Geppert, B., Rößler, F.: *Combining SDL and Pattern-Based Design for the Customization of Communication Subsystems*, in: A. Wolisz, I. Schieferdekker, A. Rennoch (Hrsg.), Formale Beschreibungstechniken für verteilte

Systeme, GMD-Studien Nr. 315, GI/ITG-Fachgespräch, Berlin, 19. -20. Juni 1997

[36] 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

[37] Geppert, B., Rößler: *Pattern-Based Reuse with SDL*, Tutorial Notes, 9th SDL Forum (SDL'99), Montreal, Kanada, 21.-25. Juni 1999

[38] Geppert, B., Rößler, F., Feldmann, R.L.: *Systematic Evolution of Reusable SDL Patterns*, Proceedings of the 2nd European Reuse Workshop (ERW'98), Madrid, Spanien, 1998

[39] Geppert, B., Rößler, F., Feldmann, R.L., Vorwieger, S.: *Combining SDL Patterns with Continuous Quality Improvement: An Experience Factory Tailored to SDL Patterns*, Proceedings of the 1st Workshop of the SDL Forum Society on SDL and MSC (SAM98), Berlin, Juni 1998

[40] Geppert, B., Rößler, F., Gotzhein, R.: *Pattern Application vs. Inheritance in SDL*, 3rd IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS'99), Florenz, Italien, Kluwer Academic Publishers, Februar 1999

[41] Geppert, B., Rößler, F., Schneider, M.: *Using SDL Patterns for the Design of a CAN-based Communication Subsystem*, Proceedings of the 8th GI/ITG-Fachgespräch Formale Beschreibungstechniken für verteilte Systeme (FBT'98), Cottbus, 1998

[42] Glässer, U., Gotzhein, R., Prinz, A.: *Towards a New Formal SDL Semantics Based on Abstract State Machines*, in: R. Dssouli, G.v. Bochmann, Y. Lahav (eds.), SDL'99 - The Next Millenium, Proc. of the 9th SDL FORUM, Elsevier Science B.V., Juli 1999

[43] Gotzhein, R., Geppert, B., Rößler, F., Schaible, P.: *Towards a New Formal SDL Semantics*, Workshop of the SDL Forum Society on SDL and MSC (SAM98), Berlin, 29.6.-1.7.98

[44] 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

[45] Gotzhein, R., Kronenburg, M., Peper, C.: *Pattern-Based Requirements Capture Applied: The SFB 501 Case Study*, in: E. Börger, B. Hörger, D. Parnas, D.

Rombach (Hrsg.), Requirements Capture, Documentation, and Validation, Dagstuhl-Seminar-Report 242, Juni 1999

[46] Kronenburg, M., Peper, C.: *Definition and Instantiation of a Reference Model for Problem Specifications*, 11th International Conference on Software Engineering and Knowledge Engineering (SEKE'99), Kaiserslautern, 1999

[47] Peper, C.: *Transformations in Pattern-Based System Specifications*, 9. GI/ITG-Fachgespräch Formale Beschreibungstechniken für verteilte Systems (FBT'99), München, Herbert-Utz-Verlag, 1999

[48] 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

[49] Rößler, F., Geppert, B.: *Applying Quality of Service Architectures to the Field-Bus Domain*, Proceedings of the 2nd IEEE International Workshop on Factory Communication Systems (WFCS'97), Barcelona, Spain, 1997

[50] 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

[51] Rößler, F., Kühlmeyer, A.: *Implementing Real-Time Communication on a Token-Ring Network*. Proceedings of the 6th Open Workshop on High Speed Networks (HSN'97), Stuttgart, October, 1997

[52] Schaible, P., Gotzhein, R.: *Einige Aspekte der Entwicklung maßgeschneiderter Kommunikationsmiddleware*, GI/ITG-Fachgespräch "Formale Beschreibungstechniken für verteilte Systeme (FBT'2000)", 22.-23.6.2000, Lübeck

Zeitschriften / Monografien

[53] Avenhaus, J., Gotzhein, R., Härder, T., Litz, L., Madlener, K., Nehmer, J., Richter, M., Ritter, N., Rombach, D., Schürmann, B., Zimmermann, G.: *Entwicklung großer Systeme mit generischen Methoden - Eine Übersicht über den Sonderforchungsbereich 501*, Informatik, Forschung und Entwicklung, 13(4):227–234, Dezember 1998

[54] Börger, E., Gotzhein, R. (Guest Editors): *Requirements Engineering: The Light Control Case Study*, Special Issue of the Journal of Universal Computer Science, Springer, to appear 2000

[55] Gotzhein, R., Bredereke, J. (Guest Editors): *Trends in Formal Description Techniques*, Theme Issue of Computer Networks and ISDN Systems, Vol. 30 (1998), Numbers 9-10, May 1998

[56] 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

[57] 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)

Industriemessen und Demonstrationen

[58] Gotzhein, R., Geppert, B., Peper, C., Rößler, F., Schaible, P.: *Ingenieurmäßige Entwicklung maßgeschneiderter Kommunikationssoftware*, CeBIT 98, Hannover, 19.-25.3.1998

[59] Gotzhein, R., Geppert, B., Peper, C., Rößler, F., Schaible, P.: *Maßschneiderung echtzeitfähiger Kommunikationssoftware*, CeBIT 98, Hannover, 19-25.3.1998

[60] Gotzhein, R., Geppert, B., Rößler, F.: *Advanced Software Reuse through Formal Methods*, Technologietransferkongress "IKT-Anwendungen auf Netzen" im Rahmen des III. Mannheimer Unternehmerforums, 6./7. Oktober 1997

[61] Feldmann, R. L., Vorwieger, S., Geppert, B., Rößler, F.: *An Experience Base Tailored to SDL Patterns* (Tool Demonstration), 1st Workshop of the SDL Forum Society on SDL and MSC (SAM98), Berlin, 29.6.- 1.7.1998

Programmkomitees mit besonderem SFB-Bezug

[62] Gotzhein, R.: Steering Committee Member des Internationalen Workshops "*SDL and MSC (SAM98)*" in Berlin, 29.6.-1.7.98; Organisation und Leitung der Sitzungen "Patterns and SDL"

6 Fremdveröffentlichungen

[63] V. R. Basili, G. Caldiera, D. Rombach: *The Experience Factory*, in: J. J. Marciniak (ed.), Encyclopedia of Software Engineering, Volume I, John Wiley & Sons, 1994, pp. 469-476

[64] T. Schmidt-Samoa: *FoReST - Entwurf und Implementierung einer Umgebung zur Erstellung formaler Systemanforderungsbeschreibungen*, Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999

[65] P. Zave, M. Jackson: *Four Dark Corner of Requirements Engineering*, ACM Transactions on Software Engineering and Methodology 6(1), 1997, pp. 1-30

Teilprojekt B5

GeneSys - Generische Systemsoftware

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

Mitarbeiter

Ergänzungsausstattung:

Dipl. Inform. L. Baum
Dipl. Inform. M. Becker

Grundausstattung:

M. Sc. V. Tamara

1 Kurzfassung der Aufgabenstellung

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.

2 Entwicklung des Teilprojekts in der Förderperiode

Aufgrund der Annahme eines Rufes an die Universität Trier kurz vor Beginn der Förderungsperiode schied Herr Dr. Sturm aus der Leitung des Teilprojekts aus.

3 Durchgeführte Arbeiten und Ergebnisse

3.1 Arbeitsschwerpunkt A: Basistechnologien für generische Systemsoftware

Im Rahmen des Arbeitsschwerpunktes A wurden in der vergangenen Förderungsperiode die zur Komposition maßschneiderbarer Laufzeitplattformen erforderlichen Basistechnologien untersucht. Gerade im Bereich der Gebäudeautomation besteht ein großer Maßschneiderungsbedarf aufgrund der unterschiedlichen Anforderungen an die Teilsysteme und des hohen Grades an Ressourcenausnutzung, der auch feingranulare Änderungen auf der Kodeebene erforderlich macht. Zur Steigerung des Wiederverwendungsgrades wurde das Konzept der generischen Komponenten angewandt. Hierbei handelt es sich um Softwarebausteine, deren Eigenschaften in begrenztem Maße über sogenannte generische Parameter in Kombination mit entsprechenden

Generatortechniken angepasst werden können, ohne notwendigerweise manuelle Änderungen am Komponentenkode vornehmen zu müssen. Die im Rahmen dieses Arbeitsschwerpunktes durchgeführten Arbeiten lassen sich grob zwei Aufgabenbereichen zuordnen: der Entwicklung der erforderlichen Basistechnologien zur Realisierung generischer Komponenten und dem Einsatz dieser Technologien bei der Fertigung von maßschneiderbaren Laufzeitplattform-Komponenten.

Basistechnologien generischer Komponenten

Die in der vergangenen Förderungsperiode geleisteten Entwicklungsarbeiten basieren auf dem Konzept der generischen Komponenten [22]. Herkömmliche, komponentenbasierte Softwareentwicklung ist im Umfeld stringenter Ressourcenanforderungen meistens nicht anwendbar, da die Komponenten entweder zu speziell sind, um sie sinnvoll wiederzuverwenden, oder aber wegen ihrer Allgemeinheit zu viele Ressourcen verbrauchen. Mit generischen Komponeten wird diesem Mißstand begegnet, indem über generische Parameter eine Maßschneiderung zentraler Komponenteneigenschaften ermöglicht wird. Hierbei können drei verschiedene Parametertypen identifiziert werden: Selektions-, Generierungs- und Kodeparameter. Selektionsparameter wählen aus einer Menge von vorgefertigten Varianten aus. Über Generierungsparameter werden entsprechende Generatoren gesteuert, welche abhängig von den aktuellen Parameterbelegungen Varianten erzeugen und diese in vorgesehene Lücken einfügen. Schließlich geben Kodeparameter überall dort, wo hochgradig individuelle Anpassungen erforderlich sind, Hilfestellungen beim manuellen Ausfüllen von belassenen Lücken.

Bei der internen Umsetzung von generischen Parametern und der Bereitstellung von zugehörigen Generatoren können verschiedene Techniken eingesetzt werden. In einem ersten Ansatz wurden hierfür der C-Präprozessor und geeignete programmiersprachliche Konzepte, wie z.B. Objekt-Orientierung oder Templates, verwendet [6]. Hiermit lassen sich Selektions- und teilweise Generierungsparameter recht effektiv realisieren. Zur weitergehenden Unterstützung von Generierungsparametern wurde anhand des sed-Editors [31] untersucht, wie Standardwerkzeuge zur Erzeugung von speziellen Generatoren genutzt werden können [22]. Über entsprechende Markierungen im Kode war es möglich, die notwendigen Transformationen durch sed-Skripte zu steuern und somit relativ einfach mächtige Generatoren zu erstellen. Während sich die Unterstützung für Kodeparameter bislang nur auf entsprechend kommentierte Lücken beschränkte, gewinnt man auch hier mit dem im Rahmen des Teilprojektes entwickelten Ansatz der Java-Metaprogrammierung einen hohen Grad an Flexibilität. Komponentenbeschreibungen können dadurch zu aktiven Dokumenten gemacht werden, die selbständig die erforderlichen Generatoren aufrufen und vielfältige Unterstützung beim Ausfüllen von belassenen Lücken leisten. Ein ähnlich flexibler, jedoch mit geringerem Aufwand verbundener Ansatz beruht auf dem Einsatz der weitverbreiteten XML-Beschreibungssprache [26] und verwandter Transformationstechniken [1]. Wie beim sed-Ansatz werden hierbei Markierungen in den Komponentenbeschreibungen eingefügt, die externe, frei gestaltbare Generatoren steuern. Die ersten Erfahrungen mit diesem vielversprechenden Ansatz legen weiterführende systematische Untersuchungen im Rahmen der kommenden Förderungsperiode nahe.

Fertigung maßschneiderbarer Laufzeitplattform-Komponenten

Die Fertigung eigener maßschneiderbarer Laufzeitplattform-Komponenten erfolgte im Hinblick auf die Evaluierung der entwickelten Basistechnologien zur Unterstützung feingranularer Maßschneiderung. Um den Realisierungsaufwand für Laufzeitplattform-Komponenten gering zu halten, wurde in einem ersten Ansatz versucht, frei verfügbare Mikrokerne als Grundlage für höhere Systemdienste einzusetzen. In [5] wurde eine Vielzahl verschiedener, frei verfügbarer Mikrokerne wie L4 [28], RTEMS [30] und RTX [29] hinsichtlich ihrer Architektur, Funktionalität und Tauglichkeit für die Arbeiten im Kontext des B5-Teilprojektes untersucht. Aufgrund fehlender Unterstützung von Echtzeitfähigkeit, übergroßen Funktionsumfangs und beschränkter Hardwareunterstützung erwiesen sich diese Kerne allerdings als ungeeignet für die angestrebte Bereitstellung maßschneiderbarer Laufzeitplattformen. Erschwerend kommt hinzu, dass ein Betriebssystemkern wesentlich die Architektur der gesamten Laufzeitplattform mitbestimmt und insbesondere auch im Kern erhebliches Maßschneiderungspotential vorhanden ist, das bei diesem Ansatz nur schwer zu nutzen wäre.

Der daraufhin durchgeführten Entwicklung eigener generischen Komponenten ging eine umfangreiche Untersuchung von Varianten elementarer Betriebssystemkonzepte und deren Abhängigkeiten untereinander voran. Im Rahmen einer simulativen Laufzeitplattform SimKernel [6] wurden hierfür 15 variable Aspekte betrachtet, wie z.B. verschiedene Varianten von Schedulingverfahren, Synchronisations- und Kommunikationsmechanismen, Speicher- und Threadmanagementstrategien. Die dabei gesammelten Erfahrungen über die Elementarvarianten und deren Abhängigkeiten wurden mit Hilfe der Design-Space-Technik [27, 24] modelliert, um später den Einsatz und die Weiterentwicklung der Varianten unterstützen zu können. Aufbauend auf den so gewonnenen Erkenntnissen wurden in [8] Elementarvarianten für die auf der Intel x86-Architektur ausführbare RTK-Laufzeitplattform erstellt, die im Wesentlichen das Scheduling und die Synchronisation von Threads übernehmen. RTK wurde dabei speziell für den Einsatz in tief eingebetteten Systemen konzipiert und bietet eine für solche Systeme hinreichende Funktionalität. Zur Quantifizierung der Laufzeit- und Speichereffizienz der einzelnen Varianten wurden verschiedene Ausprägungen der RTK-Plattform erstellt und analysiert. Die hieraus gewonnenen Resultate lieferten eine wichtige Entscheidungsgrundlage für die Maßschneiderung der RTK-Plattform.

Basierend auf den gesammelten Erfahrungen und einer gezielten Analyse der gemeinsamen und unterschiedlichen Bestandteile der untersuchten Elementarvarianten wurden nachfolgend generische Komponenten gefertigt. Hierbei kamen die entwickelten Basistechnologien bei der Realisierung der erforderlichen Generizität zum Einsatz und konnten auf ihre Tauglichkeit hin untersucht und weiterentwickelt werden. Als Resultat dieser Integrationsarbeiten entstand unter anderem die gc-RTK-Plattform, dem vollständig auf generischen Komponenten basierenden Pendant zu RTK, welche dem SFB für die Entwicklung tief eingebetteter Systeme zur Verfügung gestellt wird. Diese Plattform nimmt aber auch eine wichtige Stellung innerhalb des Teilprojekt selbst ein, da sie die Grundlage für Analysen von Basistechnologien bildet und künftig zum Gegenstand der anwendungsgestützten Konfiguration werden soll.

3.2 Arbeitsschwerpunkt B: Anwendungsgesteuerte Komposition maßgeschneiderter Systemsoftware

Ziel des Arbeitsschwerpunktes B war es, aufbauend auf dem Konzept generischer Softwarekomponenten Verfahren zur anwendungsgesteuerten Komposition maßgeschneiderter Systemsoftware zu entwickeln. Mit Hilfe geeigneter Werkzeuge sollte es Anwendungsentwicklern mit grundlegenden Betriebssystemkenntnissen ermöglicht werden, eine für ihre Anwendung maßgeschneiderte Laufzeitplattform herzustellen. Die Arbeiten in diesem Schwerpunkt konzentrierten sich daher hauptsächlich auf die Systematisierung sowie die methodische und werkzeugtechnische Unterstützung von drei Aufgabenbereichen: der Erfassung von Anwendungsanforderungen an Laufzeitplattformen, der Abbildung von Anforderungen auf geeignete generische Komponenten und der werkzeugunterstützten Konfiguration und Komposition dieser Komponenten zu einer maßgeschneiderten Laufzeitplattform.

Anforderungserfassung

Die Entwicklung einer maßgeschneiderten Laufzeitplattform bedingt die Abstimmung der funktionalen und nichtfunktionalen Eigenschaften der Plattform auf die individuellen Bedürfnisse einer konkreten Anwendung. Da jedoch nicht nur die Anwendung die erforderlichen Eigenschaften der Laufzeitplattform bestimmt, sondern umgekehrt die Eigenschaften der Laufzeitplattform auch zwangsläufig den Entwurf und die Implementierung der Anwendung beeinflussen, ist es erforderlich, Anwendungsentwicklung und Plattformentwicklung zu verzahnen und bereits sehr früh mit der Erfassung von Anwendungsanforderungen zu beginnen. Unter dieser Prämisse wurden zwei sich ergänzende Ansätze entwickelt, die es erlauben, Anforderungen an die Laufzeitplattform bereits während des Anwendungsentwurfs zu berücksichtigen und bis zu einem gewissen Grad automatisch zu erfassen.

Die in Anwendungsentwürfen enthaltenen Informationen implizieren in aller Regel bereits eine Reihe von Anforderungen an die jeweilige Laufzeitunterstützung. Die Anzahl modellierter aktiver Komponenten sowie deren Interaktionsbeziehungen geben beispielsweise erste Hinweise auf eine erforderliche Thread-Unterstützung und die benötigten Kommunikationsmechanismen; Eine Abschätzung des Modellierungsumfangs für Teilsysteme und Komponenten läßt darüber hinaus unter Umständen erste Rückschlüsse auf Rechenzeit- und Speicherplatzbedarf zu. Ausgehend von der für verteilte Kontrollsysteme naheliegenden Spezifikationssprache SDL [25] wurde daher in [4] ein Analysesystem entwickelt, das eine Reihe relevanter Anwendungseigenschaften automatisch aus konventionellen SDL-Modellen extrahiert. Durch Parsen der textuellen SDL-PR-Repräsentation eines Anwendungsentwurfs können somit automatisch Informationen beispielsweise über die Anzahl statisch und dynamisch anzulegender Threads, die Anzahl von Prozeduraufrufen, die Verwendung von Shared

Memory und die statische und dynamische Speicherkomplexität von Prozessen abgeleitet werden.

In [3] wurde darüber hinaus untersucht, inwieweit mit Hilfe gezielter Überspezifikation weitergehende Anforderungen an die Laufzeitplattform explizit in SDL-Anwendungsmodellen berücksichtigt werden können. Ziel war hierbei eine einheitliche Beschreibung von Entwurfsentscheidungen, beispielsweise bezüglich der Nutzung bestimmter Plattformdienste oder der Wahl von Interprozess-Kommunikationsmechanismen und Schedulingstrategien, mit existierenden SDL-Sprachmitteln. Das zu diesem Zweck entwickelte Konzept basiert auf einer Kombination von SDL-Mustern und vordefinierten Block- und Signal-Bezeichnern. Unterschiedliche Interaktionsmechanismen wie Pipes, FIFOs, Message Queues oder auch Shared Memory können so durch Präfixe vor Signalnamen spezifiziert werden, während die Nutzung von Systemsoftwarediensten auf wohldefinierte Signale an die Laufzeitplattform repräsentierende Proxy-Blöcke abgebildet wird. Um die auf diese Weise explizit modellierten Anforderungen bereits für die Prototypgenerierung nutzen zu können, wurde für die SDL-Entwicklungsumgebung SDT unter Solaris eine Laufzeitbibliothek entwickelt, welche die genannten Spezifikationselemente im generierten Kode erkennt und automatisch auf die gewünschten Laufzeitplattformmechanismen umsetzt.

Abbildung von Anforderungen auf Lösungen

Um eine Menge von Anforderungen möglichst einfach auf existierende Lösungsstrukturen und wiederverwendbare generische Komponenten abbilden zu können, ist es sinnvoll, bereits die Erfassung der Anforderungen auf die verfügbaren Lösungen auszurichten [18]. Mit dem Konzept der Erweiterten Design Spaces wurde hierzu eine Technik weiterentwickelt [27, 24] und eingesetzt [7], die eine semiformale Beschreibung sowohl von Anforderungen als auch von Eigenschaften beliebiger Softwareartefakte erlaubt. Design Spaces werden von einer Menge von Dimensionen aufgespannt, welche die relevanten Charakteristika des jeweils betrachteten Artefakts repräsentieren. Die möglichen Ausprägungen bezüglich dieser Charakteristika machen die sogenannten Kategorien der jeweiligen Dimensionen aus. Während ein Design Space somit einer Art Klassifikationsschema entspricht, beschreibt eine Auswahl einzelner Kategorien in einem Design Space entweder erwünschte Eigenschaften eines gesuchten Artefakts oder tatsächlich erbrachte Leistungen eines konkreten Produkts. Zur Erfassung von Anforderungen an die in Arbeitsschwerpunkt A entwickelten bzw. eingesetzten Laufzeitplattformen wurden geeignete Anforderungs-Design-Spaces aufgestellt, welche in Form von Fragebögen die Mengen möglicher Anforderungen abdecken, die sinnvoll mit der jeweiligen Plattform erfüllt werden können. Um den Umgang mit diesen Anforderungs-Design-Spaces zu erleichtern, wurden zwei Werkzeuge entwickelt, die die Erstellung und Manipulation eines Design Spaces bzw. das Erstellen von Design-Space-Profilen [7] unterstützen.

Das Design-Space-Konzept sieht über die reinen Klassifikationsmechanismen hinaus auch die Möglichkeit vor, Abhängigkeiten zwischen Kategorien und damit zwischen einzelnen Eigenschaften betrachteter Artefakte durch Angabe gewichteter Korrelationen zu dokumentieren. So lassen sich beispielsweise automatisch überprüfbare Konsistenzbedingungen für Anforderungen oder Abhängigkeiten zwischen Konfigurationsparametern definieren. Insbesondere können auch Korrelationen zwischen Anforderungs-Design-Spaces und solchen Design-Spaces spezifiziert werden, die wiederverwendbare generische Komponenten beschreiben. Auf diese Weise kann Expertenwissen über sinnvolle Zusammenhänge zwischen bestimmten Anforderungen und passenden Lösungen in Form geeignet ausgewählter und konfigurierter Komponenten explizit festgehalten und einer werkzeuggestützten Verarbeitung zugänglich gemacht werden.

Konfigurationswerkzeuge

Zur systematischen Umsetzung dieser Möglichkeiten in einen teilautomatischen Auswahl- und Konfigurationsvorgang wurde zunächst ein umfassendes Konzept [19] basierend auf einer Menge über Korrelationen verknüpfter Design Spaces erarbeitet, welches unter anderem auch ein entsprechendes Vorgehensmodell umfaßt. Demgemäß nimmt der Anwendungsentwickler zunächst eine Charakterisierung seiner Anwendung anhand eines Anwendungs-Design-Spaces vor. Da dieser Design-Space im Vokabular der Anwendungsentwicklung gehalten ist, sind hierzu keine speziellen Betriebssystemkenntnisse erforderlich. Auf Erfahrungswerten beruhende Korrelationen bilden diese Charakterisierung der Anwendung dann auf ein Anforderungsprofil in einem Anforderungs-Design-Space ab, welches ggf. an den Stellen, an denen noch keine Erfahrungswerte bezüglich geeigneter Abbildungen vorliegen, vom Anwendungsentwickler zu ergänzen ist. Ausgehend von diesem Anforderungprofil bilden Korrelationen die Anforderungen weiter auf eine geeignete Plattformarchitektur, und anschließend auf entsprechende Anforderungen an einzelne Komponenten der gewählten Plattformarchitektur ab. Diese Komponentenanforderungen ermöglichen schließlich eine teilautomatische Auswahl konkreter Komponentenimplementierungen und die Belegung der generischen Komponentenparameter. Eine erste Umsetzung dieses Konzepts in ein Unterstützungwerkzeug [7] wurde auf Basis des Betriebssystems QNX [32] vorgenommen. Im Zuge einer iterativen und interaktiven Eingabe von Anforderungen an die Laufzeitplattform QNX übernimmt besagtes Werkzeug automatisch die Auswertung aller Design-Space-Korrelationen und wählt — soweit per geeigneter Korrelationen abgedeckt — selbständig die erforderlichen QNX-Komponenten und deren Konfigurationsparameter.

3.3 Arbeitsschwerpunkt C: Experimente im Anwendungsfeld Gebäudeautomation

Ziel des Arbeitsschwerpunktes C war die kontrollierte Sammlung von Daten bezüglich der Vor- und Nachteile der in den Schwerpunkten A und B entwickelten Methoden und Werkzeuge. Dabei galt es, sowohl die Verwendbarkeit generischer Betriebssystemkomponenten für die effiziente Entwicklung maßgeschneiderter Laufzeitplattformen zu überprüfen, wie auch die Verfahren und Werkzeuge zur Unterstützung des Anwendungsentwicklers während der Maßschneiderung einer Laufzeitplattform zu validieren. Die Durchführung kontrollierter Experimente mit aussagekräftiger Vergleichsgrundlage erwies sich hierbei aufgrund einer unzureichenden Menge an Testentwicklern und mangelnder Vergleichbarkeit der Ausgangsbedingungen als unmöglich. Nichtsdestotrotz konnten im Rahmen von Fallstudien die Annahmen bezüglich einer Reduktion von Entwicklungsaufwand und Fehlerhäufigkeit sowie der Erfüllbarkeit nichtfunktionaler Anforderungen bestätigt werden.

Eine derartige Fallstudie [10] wurde in enger Zusammenarbeit mit dem Teilprojekt B4 durchgeführt, um die Auswirkungen des Einsatzes generischer Betriebssystemkomponenten wie auch generischer Kommunikationssysteme im Umfeld von Gebäudeautomationssystemen zu untersuchen. Auf Basis der generischen Komponenten der gc-RTK-Plattform wurde hierzu eine Laufzeitunterstützung für ein verteiltes, funktionell jedoch stark vereinfachtes Gebäudekontrollsystem entwickelt, wobei die erforderlichen Kommunikationsmechanismen aus dem Teilprojekt B4 integriert wurden. Anschließend wurde eine weitgehend äquivalente Systemunterstützung mit dem kommerziellen Betriebssystem QNX erstellt. Ein Vergleich der beiden Ansätze ergab einen ähnlichen Zeitaufwand für die Bereitstellung der Plattformen (zu erklären durch die manuelle Zusammenstellung der RTK-Komponenten ohne jegliche Werkzeugunterstützung) und deutliche Vorteile der RTK-Plattform gegenüber QNX bezüglich der erzielbaren Laufzeit- und insbesondere Speichereffizienz. Abgesehen davon zeigte diese Fallstudie auch erfolgreich die nahtlose Integration von Produkten der Teilprojekte B4 und B5.

Eine zweite Fragestellung betraf die mit Hilfe von Unterstützungswerkzeugen erzielbaren Effekte in Bezug auf Entwicklungsaufwand und Fehleranfälligkeit bei der Maßschneiderung von Laufzeitplattformen. Hierzu wurden mehrere QNX-Laufzeitplattformen für ausgewählte Anwendungsszenarien aus dem Bereich der Gebäudeautomation maßgeschneidert. Im Rahmen der Arbeiten zu dem Design-Space-basierten Konfigurationswerkzeug für QNX [7] wurden die Entwicklungen zum einen mit Hilfe des Konfigurationswerkzeuges, zum anderen manuell vorgenommen. Insbesondere die automatische Konsistenzprüfung, die Leitung des Anwenders durch die Konfiguration in einer bestimmten Reihenfolge und die automatische Vervollständigung der Konfiguration auf Basis von Heuristiken trugen letztlich zu einer deutlich verkürzten Entwicklungszeit und einer leicht verringerten Fehleranzahl im Falle der werkzeuggestützten Maßschneiderung bei.

4 Zusammenarbeit mit anderen Teilprojekten

Mit dem Teilprojekt B4 bestand ein enger Erfahrungsaustausch betreffend die Parametrisierung von Kommunikations- und Betriebssystemdiensten sowie die Gestaltung geeigneter Schnittstellen, der Einfluß auf die Erarbeitung der entsprechenden Techniken in den Arbeitsschwerpunkten A und B hatte. Der bereits in der ersten Förderungsperiode begonnene Abgleich der eingesetzten Techniken und Schnittstellen erwies sich hierbei als erfolgreich: so konnte im Rahmen der im Arbeitsschwerpunkt C durchgeführten Fallstudie die einfache Integrierbarkeit der Produkte aus den Teilprojekten B4 und B5 gezeigt werden.

Die Formalisierung von Komponentenabhängigkeiten und Heuristiken zur Abbildung von Anforderungen auf Lösungen wurde in Zusammenarbeit mit dem Teilprojekt C1 vorgenommen. Ferner wurden mit dem Teilprojekt C1 erste vielversprechende Ansätze für eine Formalisierung der Design-Space-Technik diskutiert, die eine weitergehende Unterstützung bei der automatischen Erkennung von Inkonsistenzen in Anforderungen und der Auswertung von Entwurfsregeln ermöglichen würde.

Bei dem Entwurf eines komplexen Gebäudeautomationssystems im Rahmen des Querschnittsprojekts Q2 übernahm das Teilprojekt beratende Funktionen in Hinblick auf die Wahl einer geeigneten Laufzeitplattform und die Nutzung angebotener Laufzeitplattformdienste.

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] M. Lösch: *Beschreibung generischer Komponenten mit XML*, AG Systemsoftware, Universität Kaiserslautern, 2000

[2] M. Näher: *Das generische Potenzial von LINUX*, AG Systemsoftware, Universität Kaiserslautern, 2000

[3] E. Becher: *Modellierung von Betriebssystemanforderungen in SDL mit SDT*, AG Systemsoftware, Universität Kaiserslautern, 1998

[4] T. See: *Erfassung von BS-Anforderungen aus SDL-Entwürfen*, AG Systemsoftware, Universität Kaiserslautern, 1998

[5] A. Schank: *Evaluierung von Mikrokernen als Basis der Entwicklung generischer Systemsoftware*, AG Systemsoftware, Universität Kaiserslautern, 1998

Diplomarbeiten

[6] A. Eckert: *Modellierung und werkzeuggestützte Verarbeitung von Parameterabhängigkeiten bei generischen Komponenten*; AG Systemsoftware, Universität Kaiserslautern, 1999

[7] A. Gilbert: *Design-Space-basierte Konfiguration des Betriebssystems QNX*; AG Systemsoftware, Universität Kaiserslautern, 1999

[8] M. Iannizzi: *Exemplarische Realisierung eines Scheduler-Subsystems für einen Realzeit-Betriebssystemkern als generische Komponente*, AG Systemsoftware, Universität Kaiserslautern, 1998

[9] T. Leyer: *Ein Konfigurationswerkzeug für generische Komponenten am Beispiel von QNX-Modulen*, AG Systemsoftware, Universität Kaiserslautern, 1998

SFB-Berichte

[10] L. Baum, M. Becker, L. Geyer, G. Molter, P. Schaible: *Integration generischer Ansätze der Kommunikationssystem- und Laufzeitplattformentwicklung*, SFB 501 Report 09/00, 2000

[11] T. Kramp, M. Adrian, and R. Koster: *An Open Framework for Real-time Scheduling Simulation*, SFB 501 Report 01/00, 2000

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

[13] T. Kramp: *Free Jazz: An User-Level Real-Time Threads Package Designed for Flexibility*, SFB 501 Report 9/98, 1998

Vorträge

[14] M. Becker: *Generative Maßschneiderung von Betriebssystemen*, GI-Fachtagung, Fachgruppe Betriebssysteme, Erlangen, März 2000

[15] J. Nehmer: *Trends in der Betriebssystemforschung*, Technische Universität München, Januar 2000

[16] L. Baum: *GeneSys – Ein Überblick,* Heinz Nixdorf Institut, Universität Paderborn, Mai 1999

[17] L. Baum: *Generierung maßgeschneiderter Varianten aus vorgefertigten Betriebssystemkomponenten,* Reuse-Workshop des SFB 501, Februar 1999

Tagungsbeiträge

[18] L. Baum, M. Becker, L. Geyer, A. Gilbert, G. Molter, V. Tamara: *Supporting Component-Based Software Development Using Domain Knowledge*, Int'l Conference on Systemics, Cybernetics and Informatics (SCI), special event on Generative and Component-Based Software Engineering, Orlando, USA, Juli 2000

[19] 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, Juni 2000

[20] J. Nehmer: *Software Reuse: Potential, Concepts, Risks*, Proc. of the IEEE Workshop on System Design Automation (SDA'2000), Rathen, März 2000

[21] 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, USA, Juni 1999

[22] 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, Juni 1999

[23] 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

[24] 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

6 Fremdveröffentlichungen

[25] R. Braek; O. Haugen: *Engineering Real Time Systems — An Object Oriented Methodology using SDL*, Prentice-Hall, 1993

[26] S. Holzner: *XML Complete*, McGraw-Hill, 1998

[27] T. G. Lane: *Studying Software Architecture Through Design Spaces and Rules*, Technical Report CMU/SEI-90-TR-18, Carnegie Mellon Univ., 1990

[28] J. Liedke: *On microkernel construction*, Proc. of the 15th ACM Symposium on Operating System Principles (SOSP), ACM Press, 1995

[29] Real Time Executives, Inc.: *The RTX Real-Time Kernel,* Real Time Executives Inc., http://www.rtx.com

[30] *Real Time Executive for Multiprocessor Systems* — Release 3.5.1 for Intel i386 C, ftp://lancelot.gcs.redstone/pub/rtems

[31] W. R. Stevens: *Advanced Programming in the UNIX Environment*, Addison-Wesley, 1992

[32] QNX Software Systems Ltd.: *QNX System Architecture*, System Documentation, QNX Software Systems Ltd., Canada, 1993

Teilprojekt B10

Anwendungsentwicklung mit vorkonfektionierten Systemen

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

Mitarbeiter

Ergänzungsausstattung:

Dipl. Inform. L. Geyer

Grundausstattung:

Dipl. Inform. G. Molter

1 Kurzfassung der Aufgabenstellung

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:

a) Bereitstellung eines Technologierahmens für wiederverwendungsorientierte, architekturbasierte Softwareentwicklung

b) Erarbeitung eines Prozessmodells für architekturzentrierte Softwareentwicklung, das auf systematische Wiederverwendung unterschiedlicher Arten von Produkten ausgelegt ist (z.B. Modellfragmente und Softwarekomponenten)

c) Entwicklung von Mess- und Analysemethoden zur Überwachung der Anwendungsentwicklung unter Ausnutzung wiederverwendbaren Wissens

d) Unterstützung der Entwicklung vorkonfektionierter Systeme für noch nicht erschlossene Domänen, sowie der Wartung und Weiterentwicklung existierender vorkonfektionierter Systeme

In der ersten Förderungsperiode wurden insbesondere die Grundlagen zu a) und b) gelegt, sowie Untersuchungen bezüglich der mit vorkonfektionierten Systemen erreichbaren Wiederverwendbarkeit angestellt. Unter Anwendung und Erweiterung der erarbeiteten Techniken und Resultate soll in der zweiten Förderungsperiode die Softwareentwicklung mit vorkonfektionierten Systemen methodisch stärker unterstützt werden; weitere Schwerpunkte werden die Neuentwicklung vorkonfektionierter Systeme sowie die Rückkopplung von Verbesserungen und Erweiterungen sein.

2 Entwicklung des Teilprojekts in der Förderperiode

Vor Anlaufen der Förderung dieses Teilprojekts hat Dr. Sturm einen Ruf an die Universität Trier angenommen und ist von diesem Zeitpunkt an aus der Leitung des Teilprojekts ausgeschieden.

3 Durchgeführte Arbeiten und Ergebnisse

3.1 Arbeitsschwerpunkt A: Konzeption und Realisierung vorkonfektionierter Softwaresysteme

Im ersten Arbeitsschwerpunkt sind in der Berichtsperiode die Grundkonzepte für die Softwareentwicklung mit vorkonfektionierten Systemen untersucht worden. Dazu wurde zunächst ein Strukturierungskonzept entwickelt, das die offene Rahmenstruktur solcher Systeme umsetzt, die zum einen geeignete wiederverwendbare Artefakte enthalten, aber auch wohldefinierte Lücken für anwendungsspezifische Funktionalität. Das einzelne vorkonfektionierte System als Instanz dieses Rahmens unterstützt dabei die gezielte Vervollständigung der definierten Lücken, sowie einen nahtlosen Übergang von den Abstraktionen des jeweiligen Anwendungsfeldes zu den Elementen des vorkonfektionierten Systems.

Architektur

Als zentrales Konzept zum Erreichen dieser Ziele hat sich in der vergangenen Förderungsperiode der Begriff der Software-Architektur herausgestellt. Im Teilprojekt B10 wird dabei ein sichtenbasierter Ansatz zur Architekturbeschreibung zurundegelegt. Die Architektur eines Systems wird durch eine Reihe spezifischer Sichten und Strukturen repräsentiert, die jeweils bestimmte Aspekte des Systems herausprojizieren und so deren gezielte Untersuchung ermöglichen.

Wie in [13] beschrieben, ist Architektur gerade im Umfeld der Entwicklung komplexer Systeme eine hohe Bedeutung beizumessen. Zum einen beeinflusst die Struktur eines Systems in hohem Maße seine Eigenschaften, insbesondere nichtfunktionale Eigenschaften wie Skalierbarkeit und Fehlertoleranz; zum anderen stellt die Architektur als Abstraktion des entwickelten Systems ein geeignetes Mittel zur Berherrschung seiner Komplexität dar.

Desweiteren kommt Architektur eine Schlüsselrolle für jegliche Produkt-Wiederverwendung zu [8][13][14]. Architekturelle Kompatibilität der wiederverwendeten Artefakte mit den bereits erstellten Strukturen und Systembestandteilen ist eine notwendige Voraussetzung für ihre erfolgreiche Integration. Diese Kompatibilität zielt beispielsweise auf Aspekte wie die zugrundeliegenden Konzepte und Interaktionsmechanismen ab, aber auch auf die Auslegung des Artefakts im Hinblick auf bestimmte nichtfunktionale Eigenschaften. Architekturelles Wissen beschreibt damit einerseits Anforderungen an die zu verwendenden wiederverwendbaren Artefakte, die bei der Auswahl eines Produkts berücksichtigt werden müssen, andererseits wird diese Selektionsaufgabe dadurch aber vereinfacht, da die Menge potentiell passender Kandidaten für die Wiederverwendung verkleinert wird.

Vorkonfektionierte Systeme

Wegen ihrer hohen Bedeutung für die erfolgreiche Realisierung von Softwareprojekten stellen Architekturen ihrerseits wertvolles wiederverwendbares Wissen dar. Im Konzept vorkonfektionierter Systeme werden daher Architektur-Wiederverwendung, architekturbasierte Softwareentwicklung und Produkt-Wiederverwendung kombiniert.

Wie in [13] und [14] beschrieben, kann ein vorkonfektioniertes System als noch unvollständige Anwendung angesehen werden. Es beinhaltet eine explizite Beschreibung der zugrundeliegenden Architektur sowie eine Auswahl dazu kompatibler wiederverwendbarer Produkte, von den grundlegenden Abstraktionen und Strukturen des Anwendungsfeldes bis hin zu Softwarekomponenten. Die Architektur stellt dabei die Grundlage für die Realisierung einer spezifischen Anwendung dar, ein Skelett, in das sowohl die im System enthaltenen wiederverwendbaren Produkte als auch neu zu realisierende, anwendungsspezifische Funktionalität eingefügt werden können.

In der vergangenen Förderungsperiode wurden im Teilprojekt B10 vorkonfektionierte Systeme für die Anwendungsgebiete Gebäudeautomation [1][2] und Multimedia-Server [4] realisiert. Anhand dieser Systeme wurden verschiedene für das Konzept vorkonfektionierter Systeme grundlegende Aspekte betrachtet. Bei der Entwicklung der beiden Systeme für das Anwendungsgebiet Gebäudeautomation wurde besonderes Augenmerk auf die hier möglichen architekturellen Alternativen sowie auf eine weitgehende Unterstützung von Wiederverwendung bei der Anwendungsentwicklung gelegt. In beiden Systemen wurden indirekte Kommunikationsmechanismen für die interne Ablaufsteuerung verwendet, die die Integration von wiederverwendbaren und anwendungsspezifischen Anwendungsbestandteilen sehr vereinfachen, sich aber bezüglich der Einbettung anwendungsspezifischer Funktionalität unterscheiden.

Anhand des vorkonfektionierten Systems für Multimedia-Server wurde exemplarisch untersucht, wie spezielle nichtfunktionale Anforderungen (hier die Auslieferung von Multimedia-Strömen mit garantierten Qualitätsmerkmalen) bei der initialen Auslegung des Systems berücksichtigt und anschließend bei der Entwicklung einzelner Anwendungen gezielt umgesetzt werden können. Zusammen mit der auch in diesem Fall geforderten Flexibilität beeinflussten diese Anforderungen wesentlich das architekturelle Design des vorkonfektionierten Systems.

Die Wiederverwendungseigenschaften dieses Systems wurden anschließend im Rahmen einer Fallstudie untersucht. Dabei wurde auf Basis des erstellten vorkonfektionierten Systems ein Server für ein komplexes, im ursprünglichen Design nicht berücksichtigtes Multimedia-Datentransfer- und Kontrollprotokoll realisiert. Die im vorkonfektionierten System vorhandenen Strukturen, Mechanismen und Produkte erwiesen sich hierbei als tragfähige Basis; beispielsweise konnte der neue Server innerhalb der dem vorkonfektionierten System zugrundeliegenden Architektur und ohne strukturelle Änderungen realisiert werden.

3.2 Arbeitsschwerpunkt B: Werkzeugunterstützung für vorkonfektionierte Softwaresysteme

Das Ziel dieses Arbeitsschwerpunktes bestand darin, geeignete Techniken, Methoden und Werkzeuge für die Anwendungsentwicklung mit vorkonfektionierten Systemen bereitzustellen. Dazu wurden zunächst bei der Entwicklung der oben erwähnten vorkonfektionierten Systeme verschiedene Beschreibungstechniken eingesetzt und evaluiert, sowie einige Techniken speziell für die Belange vorkonfektionierter Systeme weiterentwickelt. Diese Techniken wurden anschließend in ein initiales Vorgehensmodell für die Anwendungsentwicklung mit vorkonfektionierten Systemen integriert; des Weiteren wurden entsprechende Werkzeuge bereitgestellt, um ihre Anwendung zu unterstützen.

Beschreibung vorkonfektionierter Systeme

Das zentrale Strukturierungsmerkmal eines vorkonfektionierten Systems ist die ihm zugrundeliegende Architektur. Wie oben erwähnt, basiert der im Teilprojekt B10 gewählte Architektur-Beschreibungsansatz auf einem Sichtenmodell; einzelne Aspekte der Architektur werden in verschiedenen Sichten jeweils von unterschiedlichen Standpunkten aus dargestellt. Dieser Ansatz gewährleistet eine große Flexibilität für eine dem jeweiligen vorkonfektionierten System angepasste Architekturbeschreibung. Als Folge gibt es aber auch keine allgemeingültige Menge von Beschreibungstechniken, die für alle vorkonfektionierten Systeme geeignet ist. Statt dessen ist für jedes System ein eigener Satz an Beschreibungsmethoden anzugeben.

Im Folgenden sind beispielhaft einzelne Sichten und Beschreibungstechniken aufgezählt, die im Rahmen der entwickelten vorkonfektionierten Systeme Verwendung fanden.

- Architekturbeschreibungen mit der ADL Darwin [19] bieten die Möglichkeit, die Aufteilung des Gesamtsystems in Komponenten und die sie verbindenden Konnektoren zu beschreiben. Die formale Semantik der ADL lässt bereits auf dieser Abstraktionsebene gezielte Analysen zu.
- Auf niedrigeren Abstraktionsebenen wurden objektorientierte Modellierungstechniken verwendet, um das Design einzelner Komponenten zu beschreiben, beispielsweise die OMT-Methode [20] und die UML [21]. Die Vielzahl der mit UML spezifizierbaren Modelle erlaubt eine umfassende Beschreibung verschiedener Aspekte des Komponentendesigns.
- Zur Formulierung von Anforderungen bei der Anwendungsentwicklung mit vorkonfektionierten System wurden Szenarien oder Use-Cases [21] benutzt,

die eine sehr natürliche Darstellung der Anforderungen ermöglichen. Solche Szenarien können bereits während der Entwicklung des vorkonfektionierten Systems auf bestimmte Systemstrukturen abgebildet werden; dieses Wissen kann bei der Anwendungsentwicklung zur Unterstützung des architekturellen Designs genutzt werden. Eine weitere Anwendungsmöglichkeit von Szenarien ist Architekturanalyse beispielsweise in Form der erweiterten SAAM-Methode [11].

- Als weitere Technik zur Beschreibung von Anforderungen kam die Software Cost Reduction-Methode (SCR) [18] zum Einsatz. Die Anforderungsspezifikation anhand dieser Methode kann dabei durch domänenspezifische Anforderungsrahmen und -templates [12] unterstützt werden, die auf generische Weise für das jeweilige Anwendungsgebiet typische Problemstellungen beschreiben.
- Das einem vorkonfektionierten System zugrundeliegende Vokabular wurde jeweils in einem Dictionary zusammengefasst und definiert.
- Bei der Modellierung der Datenstrukturen eines Systems wurde auf Entity-Relationship-Diagramme [17] zurückgegriffen.
- Design Space-Profile [15] stellen einen weiteren Bestandteil der Beschreibung vorkonfektionierter Systeme dar. Zum einen kann auf diese Weise das generelle Anwendungsgebiet eines vorkonfektionierten Systems umrissen werden. Zum anderen können in entsprechenden Sub-Design Spaces die relevanten Charakteristika von im vorkonfektionierten System enthaltenen wiederverwendbaren Produkten festgehalten bzw. Anforderungen an noch zu realisierende anwendungsspezifische Systembestandteile ausgedrückt werden.
- Zur Beschreibung nichtfunktionaler Eigenschaften – beispielsweise der von einem vorkonfektionierten System benötigten CPU-Zeit und Kommunikationsbandbreite – wurde eine spezielle, in der AG Systemsoftware entwickelte Technik für die Modellierung von Ressourcenverfügbarkeit und -Nutzung [16] eingesetzt.

Softwareentwicklung mit vorkonfektionierten Systemen

Die Anwendungsentwicklung mit vorkonfektionierten Systemen basiert auf der Kombination der Wiederverwendung einer Architektur als Strukturierungskonzept mit dem Konzept der architekturbasierten Wiederverwendung, d.h. der Nutzung architekturellen Wissens zur Unterstützung von Produkt-Wiederverwendung. Dadurch sowie durch den im Vorfeld geplanten Einsatz wiederverwendbarer Produkte können Effektivität und Wiederverwendungsgrad bei der Durchführung von Anwendungsprojekten erhöht werden. Wie in [13] und [14] dargestellt, müssen diese Aspekte bei der Durchführung aller Entwicklungsaktivitäten explizite Berücksichtigung finden. Anders als im Falle von ad-hoc-Wiederverwendung sind Wiederverwendungsaspekte hier fester Bestandteil des Entwicklungsprozesses; sie können geplant werden und bei Kosten- und Aufwandsabschätzungen Berücksichtigung finden.

Im Teilprojekt B10 wurde in der vergangenen Förderungsperiode eine solche Vorgehensweise für die Anwendungsentwicklung mit vorkonfektionierten Systemen erarbeitet, die Wiederverwendungsaspekte und die Nutzung architekturellen Wissens explizit aufnimmt; sie wird in [12] im Überblick beschrieben. In einer internen Fallstudie wurde diese Vorgehensweise anschließend erfolgreich anhand eines einfachen vorkonfektionierten Systems für das Anwendungsgebiet Gebäudeautomation erprobt. Insbesondere durch die Bereitstellung erprobter Strukturen, die bei der Anwendungsentwicklung mit geringfügigen Änderungen übernommen werden können, wurde in dieser Fallstudie der Entwicklungsaufwand beträchtlich verringert.

Unterstützungstechniken und -werkzeuge

Die oben bereits erwähnte Design Space-Technik [15] kann im Kontext vorkonfektionierter Systeme für verschiedene Zwecke eingesetzt werden. Zum einen erlaubt sie die Überprüfung von Konsistenzkriterien für die Erfassung von Anforderungen oder für bestimmte Designentscheidungen. Zum anderen können in Form von Korrelationen im Design Space Heuristiken zur Abbildung beispielsweise von Anforderungen auf geeignete Lösungsstrukturen ausgedrückt werden.

In der vergangenen Förderungsperiode wurde im Teilprojekt B10 schwerpunktmäßig untersucht, wie die Auswahl eines geeigneten vorkonfektionierten Systems für eine bestimmte Anwendung mit Hilfe der Design Space-Technik unterstützt werden kann. Eine weitere Anwendung finden Design Spaces in späteren Entwicklungsphasen bei der Auswahl wiederverwendbarer Produkte, die vom eingesetzten vorkonfektionierten System zur Verfügung gestellt werden. Dazu wurden Design Spaces für Gebäudeautomationssysteme [1][2] sowie für Multimedia-Server [4] aufgestellt. In [10] wurde gezeigt, wie Design Spaces als Basistechnologie in einem Produktlinien-Ansatz verwendet werden können, der es bei geeigneter Werkzeugunterstützung erlauben soll, Mitglieder einer Produktfamilie automatisch zu generieren.

Bei der Verwendung von Design Spaces stellt deren inhärente Komplexität ein prinzipielles Problem dar. Zur Beherrschung dieser Komplexität wurden in der vergangenen Förderungsperiode in Zusammenarbeit mit dem Teilprojekt B5 zwei Werkzeuge entwickelt. Das erste Werkzeug, *Reboost*, erlaubt die Erstellung und Wartung von Design Spaces; es bietet die Möglichkeit, durch einen Design Space zu navigieren und entsprechende Beschreibungselemente einzufügen, bzw. den Design Space an eine neue Situation anzupassen. Das zweite Werkzeug, *D-Space-1*, wurde im Rahmen einer Diplomarbeit [3] entwickelt; es dient dazu, Belegungen in einem Design Space vorzunehmen, d.h., Anforderungsprofile sowie Profile wiederverwendbarer Produkte zu erstellen. Das Tool unterstützt den Entwickler dabei durch die automatische Überprüfung der im Design Space kodierten Abhängigkeiten.

Eine weitere im Zentrum der Arbeiten im Teilprojekt B10 stehende Technik ist die erweiterte SAAM-Analyse [9][11]. SAAM ist ein szenariobasiertes Analyseverfahren, bei dem typische Interaktionen mit einem System mit Hilfe von Szenarien beschrieben werden; bei ihrer Auswertung wird untersucht, ob und mit welchem Aufwand sie mit der vorliegenden Architektur umgesetzt werden können. Im Umfeld vorkonfektionierter Systeme wird die erweiterte SAAM-Analyse beispielsweise dafür eingesetzt, die Wiederverwendungseigenschaften sowohl eines ganzen Systems als auch einzelner wiederverwendbarer Produkte zu evaluieren sowie den Rahmen für deren Einsetzbarkeit zu dokumentieren. Desweiteren sind SAAM-Analysen fester Bestandteil von auf vorkonfektionierten Systemen basierenden Anwendungsprojekten, um die Integrität und Qualität der entwickelten Anwendung zu überwachen, sowie zur Unterstützung von Auswahl und Parametrisierung wiederverwendbarer Artefakte.

Neben den zuvor erwähnten Design Space-Werkzeugen wurden in der vergangenen Förderungsperiode in B10 zwei weitere Werkzeuge entwickelt [6]: ein datenbankgestütztes Protokollierungswerkzeug, das während der Durchführung von Fallstudien die einfache, aber dennoch exakte Protokollierung von Aufwand und entdeckten Fehlern erlaubt, sowie ein ebenfalls datenbankgestütztes Werkzeug für Erstellung und Pflege domänen- und anwendungsspezifischer Dictionaries. Ein solches Dictionary ist Bestandteil jedes vorkonfektionierten Systems; es definiert die darin vorkommende Begriffswelt sowie die Zusammenhänge zwischen diesen Begriffen. Während der Realisierung einer spezifischen Anwendung wird dieses Dictionary um projektspezifisches Wissen erweitert bzw. verfeinert; neben komfortablem Suchen und Browsing des Dictionaries werden auch diese Änderungsaktivitäten von dem realisierten Werkzeug unterstützt.

Eine Evaluierung der zuvor erstellten Werkzeuge erfolgte an Hand kleinerer Fallstudien, mit denen der Nutzen beispielsweise der Design Space-Werkzeuge bestätigt werden konnte. Durch die beiden Werkzeuge wurden Erstellung und Handhabung von Design Spaces sowie die Überprüfung der Einhaltung der Konsistenzbedingungen im Vergleich zu manueller Erstellung und Konsistenzprüfung signifikant erleichtert. Das Dictionary-Werkzeug wurde darüberhinaus bereits im Rahmen des Querschnittsprojekts Q1 [7] im SFB erfolgreich eingesetzt; dadurch konnten den beteiligten Entwicklern Definitionen aller im Entwicklungsprojekt auftretenden Begriffe und Konzepte zeitnah und konsistent zur Verfügung gestellt werden.

4 Zusammenarbeit mit anderen Teilprojekten

Bei den Arbeiten im Querschnittsprojekt Q1 wurden von den beteiligten B10-Mitarbeitern Aspekte der Wiederverwendung von Domänenwissen sowie des Konzepts der geplanten Wiederverwendung in das im Team erstellte Referenzprozessmodell eingebracht. Die Berücksichtigung dieser Gegebenheiten stellte eine Voraussetzung für die anschließende Einordnung der Arbeiten des Teilprojekts in das Referenzprozessmodell dar.

Im Querschnittsprojekt Q2 wurde in einer Fallstudie im Rahmen eines Praktikums ein komplexes Gebäudeautomationssystem entworfen und auszugsweise implementiert. Dabei waren Mitarbeiter aus dem Teilprojekt B10 an den Reviews des Systemdesigns beteiligt, um zum einen die entwickelten Architekturen zu bewerten, und zum anderen Design-Empfehlungen im Hinblick auf die Realisierung spezifischer nichtfunktionaler Eigenschaften zu geben.

Das Querschnittsprojekt Q3 wurde durchgeführt, um die in den vergangenen Förderungsperioden gesammelten Erfahrungen bezüglich der Domäne Gebäudeautomation in einem Domänenmodell zusammenzufassen. Der Beitrag des Projekts B10 zu diesem Querschnittsprojekt bestand unter anderem in der Bereitstellung von Verfahren zur Domänenanalyse; als initiales Verfahren wurde hier die Commonality Analysis des FAST-Ansatzes [22] ausgewählt. Im Laufe der Domänenanalyse wurde dieses Verfahren an die Bedürfnisse der Domäne, sowie des personellen Umfeldes angepasst. Die Ergebnisse des Projekts in Form des angepassten Domänenanalyseverfahrens und eines Modells der Domäne Gebäudeautomation sind in einem SFB-Bericht [5] zusammengefasst.

5 Eigene Veröffentlichungen im Berichtszeitraum

Diplomarbeiten

[1] S. Albus: *Eine Design Space-Klassifikation und Realisierung eines event-gesteuerten vorkonfektionierten Systems zur Gebäudeautomation*, AG Systemsoftware, Universität Kaiserslautern, 1998

[2] S. Bötzel: *Eine Design Space-Klassifikation und Realisierung eines markt-basierten vorkonfektionierten Systems zur Gebäudeautomation*, AG Systemsoftware, Universität Kaiserslautern, 1998

[3] A. Gilbert: *Design-Space-basierte Konfiguration des Betriebssystems QNX*, AG Systemsoftware, Universität Kaiserslautern, 1999

[4] F. Hüther: *Ein vorkonfektioniertes System für echtzeitfähige Multimedia-Server, 1998*

SFB-Berichte

[5] 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

[6] L. Baum, M. Becker, L. Geyer, G. Molter: *Zwei Unterstützungswerkzeuge für Experimente auf dem Gebiet domänenspezifischer Softwareentwicklung*, SFB 501 Report 14/99, Dept. of Computer Science, University of Kaiserslautern, 1999

[7] R. Feldmann, J. Münch, S. Queins, S. Vorwieger, G. Zimmermann: *Baselining a Domain-Specific Software Development Process*, SFB 501 Report 02/99, Dept. of Computer Science, University of Kaiserslautern, 1999

Vorträge

[8] J. Nehmer: *Softwarearchitektur – Ein Schlüsselkonzept zur Bewältigung der Softwarekrise*, Universität Würzburg, Juli 2000

[9] L. Geyer: *Architekturanalyse in wiederverwendungsorientierten Entwicklungsprozessen,* Reuse-Workshop des SFB 501, Februar 1999

Tagungsbeiträge

[10] L. Baum, M. Becker, L. Geyer, A. Gilbert, G. Molter, V. Tamara: *Supporting Component-Based Software Development Using Domain Knowledge*, Proc. of the 4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000, Orlando, USA, 23-26 Juli 2000

[11] G. Molter: *Integrating SAAM in Domain-Centric and Reuse-based Development Processes*, Proc. of the 2nd Nordic Workshop on Software Architecture (NOSA), Ronneby, Sweden, 12-13 August 1999

[12] L. Baum, M. Becker, L. Geyer, G. Molter: *A Process View on Architecture-Based Software Development*, Proc. of the 1st Working IFIP Conference on Software Architecture (WICSA-1), San Antonio, Texas, USA, 22-24 Feb. 1999

[13] L. Baum, M. Becker, L. Geyer, G. Molter: *The Role of Architecture for Complex Systems Development*, Proc. of the 11th. Int'l Conf. on Software Engineering and its Applications 1998, Paris, France, 8-10 Dec. 1998

[14] L. Baum, M. Becker, L. Geyer, G. Molter: *Using Software Architecture as a Catalyst for Reuse*, Proc. of European Reuse Workshop 1998, Madrid, Spain, 4-6 Nov. 1998

[15] 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

[16] 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, 28. June - 2. July 1999

6 Fremdveröffentlichungen

[17] P. P. Chen: *The Entity-Relationship Model – Towards a Unified View of Data*, ACM Transactions on Database Systems, Vol. 1, No. 1, S. 9-36, März 1976

[18] Heninger, K. L.: *Specifying software requirements for complex systems: New techniques and their application*, IEEE Transactions on Software-Engineering, SE-6(1), pp. 2-13, 1980

[19] 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, 1995, pp. 137-153

[20] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen: *Object-Oriented Modeling and Design*, Prentice Hall, 1991

[21] J. Rumbaugh, I. Jacobsen, G. Booch: *The Unified Modeling Language – Reference Manual*, Addison-Wesley, Object Technology Series, 1999

[22] D. Weiss: *Software Product-Line Engineering: A Family-Based Software Development Process*, Addison Wesley, 1999

Teilprojekt C1

Formale Beschreibungstechniken

Leiter

Prof. Dr. Jürgen Avenhaus	Prof. Dr. Klaus Madlener
geb. 19.08.1941	geb. 15.01.1947
Universität Kaiserslautern	Universität Kaiserslautern
Postfach 3049	Postfach 3049
67653 Kaiserslautern	67653 Kaiserslautern
Tel.: 0631/205-2633	Tel.: 0631/205-2268
Fax: 0631/205-2156	Fax: 0631/205-2156
avenhaus@informatik.uni-kl.de	madlener@informatik.uni-kl.de

Mitarbeiter

Ergänzungsausstattung:

Dipl. Inform. T. Deiß (bis 30.6.1999)

Dipl. Inform. M. Kronenburg (seit 1.4.1995)

Grundausstattung:

Dipl. Inform. R. Eschbach (seit 1.1.1998)

1 Kurzfassung der Aufgabenstellung

Bei der Entwicklung großer Systeme werden in zunehmendem Umfang formale Beschreibungstechniken eingesetzt. Damit werden sowohl einzelne (Teil-)Produkte als auch der Entwicklungsprozess beschrieben. Hierzu wird eine Vielzahl verschiedener Beschreibungstechniken verwendet, deren Zusammenhang jedoch noch nicht ausreichend geklärt ist.
Aufgaben des Teilprojektes C1 sind die Untersuchung formaler Beschreibungstechniken im Kontext der Anwendungsdomäne sowie die Unterstützung der übrigen Teilprojekte bei der Verwendung derartiger Techniken. Bei diesen Arbeiten werden vor allem die Aspekte Verfolgbarkeit, Generizität und Wiederverwendbarkeit betrachtet.

Konkretes Ziel für die zweite Förderperiode war, das Erstellen formaler Systembeschreibungen im Bereich von Anforderungsspezifikationen bis zu Entwürfen mit Hilfe der Konzepte Wiederverwendung und Komposition zu unterstützen. Des Weiteren sollte die Kombination verschiedener Beschreibungstechniken exemplarisch anhand der Kombination von TLA und statecharts weiter untersucht werden sowie der bereits in der ersten Förderperiode begonnene Vergleich von Beschreibungstechniken fortgesetzt werden.

2 Entwicklung des Teilprojekts in der Förderperiode

Die Entwicklung des Teilprojektes wurde durch drei Faktoren beeinflusst:

- Da im Rahmen der Ergänzungsausstattung nur eine Stelle bewilligt worden war, konnte der Arbeitsschwerpunkt B im Berichtszeitraum nicht behandelt werden.
- Bei der Bearbeitung des Arbeitsschwerpunktes A hat sich herausgestellt, dass zur Beschreibung von Familien *großer* Systeme neben geeigneten Parametrisierungstechniken vor allem auch weitere Strukturierungstechniken sowie eine Werkzeugunterstützung unbedingt erforderlich sind. Deswegen konnte der Übergang zur Entwurfsphase nicht wie geplant durchgeführt werden.
- Entsprechend seiner Koordinationsrolle bezüglich der Verwendung formaler Beschreibungstechniken im SFB ist das Teilprojekt C1 als einziges Teilprojekt an allen Querschnittsprojekten im Berichtszeitraum aktiv beteiligt gewesen. Dadurch wurde für SFB-übergreifende Themenstellungen beträchtlich mehr Zeit aufgewendet als veranschlagt.

3 Durchgeführte Arbeiten und Ergebnisse

3.1 Arbeitsschwerpunkt A: Beschreibungstechniken für Familien von Systemen

Kenntnisstand bei der letzten Antragsstellung und Ausgangsfragestellung:

Die in der ersten Förderperiode gemachten Erfahrungen hatten gezeigt, dass es erforderlich ist, Beschreibungen von Familien von Systemen, insbesondere auch auf der Ebene der Anforderungen und des Entwurfs, durch geeignete (formale) Beschreibungstechniken so zu unterstützen, dass eine Wiederverwendung der Systeme und ihrer Beschreibungen ermöglicht wird. Existierende formale Beschreibungstechniken sollten daraufhin untersucht werden, ob die zur Verfügung gestellten Möglichkeiten ausreichend sind, solche Familien zu beschreiben.
Konkret sollte die in der ersten Förderperiode bereits verwendete Temporallogik untersucht werden. Diese erlaubt zum einen eine deskriptive Beschreibung eines Systems und damit eine explizite Darstellung seiner Eigenschaften, was gerade im Bereich der Anforderungen sinnvoll und hilfreich ist. Zum anderen lagen für diese Temporallogik bereits Eigenschaftsmuster vor [13], die in Zusammenarbeit mit Teilprojekt B4 entwickelt worden waren und eine parametrisierte Beschreibung einzelner Eigenschaften eines Systems ermöglichen.

Angewandte Methoden:

Die Eignung der verwendeten Beschreibungstechniken, Methoden und Werkzeuge wurde stets anhand von Fallstudien überprüft. Die dabei gemachten Erfahrungen waren ihrerseits Ausgangspunkt entsprechender Erweiterungen. Die Fallstudien wurden zunächst in der Domäne Gebäudeautomation durchgeführt, siehe [16], [20] sowie die Ergebnisberichte zu den Querschnittsprojekten Q2 und Q4. Um die Anwendbarkeit des entwickelten Ansatzes für eine weitere Domäne im Bereich der reaktiven Systeme zu untersuchen, wurden im Teilprojekt C1 zwei Fallstudien in der Domäne Verkehrsbeeinflussung durchgeführt [2], [3].

Ergebnisse und ihre Bedeutung:

Referenzmodell für Problemspezifikationen. Bereits in der ersten Förderperiode hat sich gezeigt, dass es zunächst unbedingt erforderlich ist, den gewünschten Inhalt eines Dokuments unabhängig von einer bestimmten Beschreibungstechnik zu charakterisieren. Daher haben wir ausgehend von den Arbeiten von Jackson und Zave [21], [22] ein Referenzmodell für *Problemspezifikationen* entwickelt [15]. Demnach besteht eine Problemspezifikation aus einer Menge von *Bezeichnern* und einer Menge von *Aussagen*, die wie folgt klassifiziert werden.

Für jeden Bezeichner muss es zunächst eine Erläuterung geben, die beschreibt, welches Phänomen in der realen Welt, zum Beispiel die Temperatur in einem Raum oder die von einem Sensor gemessene Temperatur, durch einen Bezeichner repräsentiert wird. Des Weiteren wird jedem Bezeichner ein *Einflussbereich* zugeordnet. Hierzu wird ein *System*, unter dem wir einen Teil der realen Welt verstehen, unterteilt in eine

Maschine und eine *Umgebung*. Die Umgebung ist der Teil eines Systems, der bereits vorhanden ist; hierzu gehören zum Beispiel Phänomene wie die Temperatur oder Helligkeit in einem Raum oder bereits installierte Sensoren oder Aktuatoren. Im Gegensatz hierzu ist die Maschine der Teil eines Systems, der noch entwickelt werden muss. Dies ist in der Regel die zu entwickelnde Software, aber auch Sensoren oder Aktuatoren, die noch installiert werden müssen, gehören dazu. Der Einflussbereich eines Bezeichners hängt davon ab, ob das durch ihn repräsentierte Phänomen zur Maschine oder zur Umgebung gehört und ob es für den jeweils anderen Teil sichtbar ist.

Die Klassifikation der Aussagen basiert auf der Unterscheidung in *optative* und *indikative* Aussagen sowie den Einflussbereichen der in einer Aussage verwendeten Bezeichner. Indikative Aussagen beschreiben feststehende Eigenschaften des betrachteten Systems, optative Aussagen hingegen gewünschte Eigenschaften. Dieses Referenzmodell für Problemspezifikationen wurde für die betrachtete Temporallogik instanziiert.

Strukturierte Problemspezifikationen. Bereits die Erfahrungen aus der ersten Förderperiode hatten gezeigt, dass die verwendete Temporallogik zwar prinzipiell ausreichend ist, wesentliche, d. h. vor allem die funktionalen und zeitbehafteten Eigenschaften eines reaktiven Systems zu spezifizieren, dass sie aber für die Spezifikation *großer* Systeme, wo 1000, 10000 oder noch mehr Phänomene und eine in der gleichen Größenordnung liegende Anzahl von Eigenschaften betrachtet werden, praktisch nicht einsetzbar ist. Dass diese Zahlen realistisch sind, haben die durchgeführten Fallstudien gezeigt, siehe auch den Bericht zu Querschnittsprojekt Q4. Daher war es notwendig, die Temporallogik mit Strukturierungskonzepten anzureichern.

Dadurch ist es zunächst einmal möglich, eine Problemspezifikation in *Beschreibungsklassen* zu unterteilen. Diese Beschreibungsklassen können zum einen parametrisiert sein, zum anderen können sie in Aggregations- und/oder Vererbungsbeziehungen zu anderen Beschreibungsklassen stehen. Wesentliches Ziel bei diesen Arbeiten war neben einer Erhöhung der praktischen Einsetzbarkeit der verwendeten Temporallogik stets eine mathematisch präzise Definition der zu Grunde liegenden Syntax und Semantik der integrierten Strukturierungskonzepte. Die Ergebnisse dieser Arbeiten sind in [6] zusammengefasst.

Werkzeugunterstützung. Neben der Möglichkeit zur Strukturierung ist das Vorhandensein einer entsprechenden Werkzeugunterstützung eine weitere notwendige Voraussetzung zur Erstellung von Problemspezifikationen großer Systeme und damit zur Durchführung der Fallstudien gewesen. Daher wurde ein Werkzeug entwickelt [4], das zum einen die Eingabe strukturierter Problemspezifikationen wie auch das beschriebene Referenzmodell unterstützt. Zum anderen verwaltet dieses Werkzeug verschiedene Verfolgbarkeitsrelationen, die teilweise automatisch erzeugt werden.
Um nicht nur das Erstellen von Problemspezifikationen, sondern auch das Arbeiten mit ihnen zu erleichtern, erzeugt dieses Werkzeug unter anderem eine Ausgabe in Form von html-Dokumenten, bei denen die Verfolgbarkeitsrelationen auf hyper-links abgebildet werden. Dadurch ist eine komfortable Navigation auch durch umfangrei-

che Problemspezifikationen möglich, siehe zum Beispiel http://www-avenhaus.informatik.uni-kl.de/forest/EXAMPLES/JUCS/JUCSStart.html.
Eine weitere wesentliche Erfahrung aus den durchgeführten Fallstudien ist, dass zur Erhöhung der Verständlichkeit es unbedingt erforderlich ist, für die temporallogischen Spezifikationen der Eigenschaften entsprechende Übersetzungen in natürlicher Sprache anzugeben. Deswegen wird dies von dem entwickelten Werkzeug auch explizit vom Benutzer abgefragt.

Wiederverwendung. Die Anwendung des entwickelten Ansatzes in einer anderen Domäne als der der Gebäudeautomation, nämlich in der Domäne Verkehrsbeeinflussung hat gezeigt [2], [3], dass sich teilweise ganze Beschreibungsklassen, zum Beispiel solche für allgemeine Sensoren oder Aktuatoren, von der Domäne Gebäudeautomation übernehmen lassen. Außerdem konnte in beiden Domänen eine ähnliche Gruppierung der Beschreibungsklassen vorgenommen werden, die eine systematische Vorgehensweise bei der Erstellung von Problemspezifikationen unterstützt.
Insgesamt hat sich gezeigt, dass der entwickelte Ansatz nicht nur auf der Ebene einzelner Eigenschaften in Form von Anforderungsmustern ein hohes Wiederverwendungspotential besitzt, sondern dass es aufgrund der Integration der Strukturierungskonzepte auch möglich ist, größere Teile einer Spezifikation, d. h. Beschreibungsklassen, leicht abgewandelt in anderen Spezifikationen wiederzuverwenden. Bezüglich Wiederverwendung sei auch auf den Bericht zu Querschnittsprojekt Q4 verwiesen, wo dieser Aspekt ebenfalls im Vordergrund stand.

Weitere Einzelergebnisse der in der Domäne Verkehrsbeeinflussung durchgeführten Fallstudien sind ein erstes Beweismuster zum Nachweis eines wichtigen Zusammenhanges von Eigenschaften einer Problemspezifikation [2] sowie ein erster Vorschlag zur Integration tabellenbasierter Beschreibungstechniken in unseren Ansatz, um in bestimmten Fällen kompaktere Spezifikationen zu erzielen [3].

Vergleiche mit anderen Arbeiten außerhalb des SFB:

Das entwickelte Referenzmodell ist vor allem beeinflusst durch die Arbeiten von Jackson und Zave [21], [22] und teilweise auch durch die Arbeiten von Parnas et al. [23].

Die Notwendigkeit, vor allem objektorientierte Beschreibungstechniken, insbesondere UML [24], einen mathematisch präzisen semantischen Unterbau zu geben, wird von vielen Autoren betont, vgl. zum Beispiel [22], [25], [26], [27], [28]. Was die formale Erfassung des Begriffs *Objekt* anbelangt, so entspricht unser Ansatz den zu Grunde liegenden Ideen vergleichbarer Ansätze, siehe zum Beispiel Z++ [29], Object Calculus [30] oder auch der Ansatz von Ehrich [28]. Diese Ansätze unterscheiden sich von unserem darin, dass sie nicht auf die Spezifikation reaktiver Systeme zugeschnitten sind, die Konzepte Vererbung, Aggregation und Parametrisierung anders interpretieren, mehr auf operationale Repräsentationen von Lösungen als auf deskriptive Beschreibungen von Problemen abzielen oder dadurch dass die formale Erfassung der Konzepte auf Grundlage der Kathegorientheorie und nicht wie bei uns auf der Basis der Mengentheorie erfolgt. Insbesondere wird in keinem der Ansätze eine Kombination von Strukturierungskonzepten und Klassifikationen, wie sie in unserem Referenzmodell enthalten sind, vorgenommen.

Offene Fragen:

Zunächst muss die Untersuchung der Eignung des entwickelten Ansatzes anhand weiterer Fallstudien fortgesetzt werden. Dabei ist insbesondere zu klären, mit welchen Konzepten zum einen die Erstellung von Problemspezifikationen und zum anderen deren Verwendung weitergehend unterstützt werden kann.
Hinsichtlich der Erstellung ist zu untersuchen, welche semantischen Analysen sinnvoll und möglich sind und welche existierenden Werkzeuge hierfür wie eingesetzt werden können. Des Weiteren erscheint es sehr hilfreich zu sein, gezielt weitere Beschreibungstechniken, wie zum Beispiel Tabellen- oder graphische Notationen, in den Ansatz zu integrieren. Dabei ist es noch offen, wie die Integration sowohl auf syntaktischer als auch semantischer Ebene am besten erfolgen kann.
In diesem Zusammenhang muss auch geklärt werden, inwieweit weitere nicht-funktionale Eigenschaften möglichst formal erfasst werden können, da bisher vor allem funktionale und zeitbehaftete Eigenschaften betrachtet wurden.
Hinsichtlich der Verwendung von Problemspezifikationen hat es sich herausgestellt, dass häufig Projektionen einzelner Teile einer Problemspezifikationen hilfreich wären. In diesem Zusammenhang muss daher untersucht werden, welche Projektionen dies im Allgemeinen sind und wie diese konsistent und möglichst automatisch erzeugt werden können.

Weitere zentrale Fragen sind, wie das vorhandene Referenzmodell für Problemspezifikationen auf Entwurfsspezifikationen ausgedehnt werden kann und welche Verfolgbarkeitsrelationen zwischen Problemspezifikationen und Entwurfsspezifikationen hilfreich sind.

3.2 Arbeitsschwerpunkt B: Kompositionsoperatoren

Konnte wegen des in Abschnitt 2 angegebenen Grundes nicht bearbeitet werden.

3.3 Arbeitsschwerpunkt C: Verknüpfung und Bewertung von Beschreibungstechniken

Kenntnisstand bei der letzten Antragsstellung und Ausgangsfragestellung:

Hinsichtlich der Kombination von Beschreibungstechniken lagen erste Ergebnisse zur Kombination einer eigenschaftsorientierten Beschreibungstechnik (TLA) und einer modellorientierten Beschreibungstechnik (statecharts) vor. Insbesondere waren die Syntax einer solchen Kombination und die prinzipielle Methode zur Festlegung der Semantik einer Erweiterung von statecharts um TLA Konstrukte (TLA Formeln beschreiben die Semantik der erweiterten statecharts) bereits erprobt.

Die Eignung der kombinierten Beschreibungstechnik für die Spezifikation reaktiver Systeme mit Zeitbedingungen sollte bewertet werden, eine Methodologie zur Erstellung solcher Beschreibungen sollte erarbeitet und praktiziert werden, sowie allgemeine Methoden zur Erweiterung und Kombination von formalen Beschreibungstechniken untersucht werden.

Bezüglich der Bewertung von Beschreibungstechniken gab es einen (informellen) Kriterienkatalog, der erweitert und formalisiert werden sollte.

Angewandte Methoden:

Kombination von Beschreibungstechniken. Sowohl TLA als auch statecharts werden zur Beschreibung des Verhaltens eines reaktiven Systems verwendet. Zur Definition der Semantik einer Kombination dieser Beschreibungstechniken wurde eine Abbildung in die semantische Begriffswelt von TLA verwendet. Diese erlaubt es, sowohl die etwas "knifflige" Semantik von statecharts genau zu erfassen als auch die in TLA verfügbare Begriffswelt, wie zum Beispiel den Verfeinerungsbegriff, zu verwenden bzw. zu übertragen. Die syntaktischen Konstrukte der kombinierten Sprache sollten so weit wie möglich natürlich, d. h., ihre Wirkung leicht verständlich und überschaubar sein. Hierfür sollte die Analyse der Fallstudien Liftsystem [17], Ampelanlage [5], sowie Produktionszelle [1] zur Auswahl sowie zum Vorschlag einer Methodologie dienen.

Bewertung von Beschreibungstechniken. Als Grundlage für eine mehr formale Klassifikation und Bewertung von Beschreibungstechniken wurde für eine eingeschränkte Klasse formaler Beschreibungstechniken, die für die Spezifikation reaktiver Systeme geeignet sind, ein allgemeiner, formaler Rahmen definiert. Die Eignung des Rahmens wurde nachgewiesen, indem exemplarisch gezeigt wurde, dass verschiedene Temporallogiken Instanzen dieses Rahmens sind ([7], [8]).

Ergebnisse und ihre Bedeutung:

Kombination von Beschreibungstechniken. Die Ergebnisse sind im Wesentlichen in der Dissertation [5] sowie in der Arbeit [17] dargestellt. Sie zeigen, dass der vorgeschlagene Weg zur Kombination von formalen Beschreibungstechniken möglich ist und auch Erfolg versprechend für die Erweiterung bzw. Kombination anderer Beschreibungstechniken ist. Die Vorgehensweise zur Erstellung solcher kombinierten Beschreibungen sieht vor, dass geeignete Beschreibungen der Maschine und der Umgebung sowie der optativen und indikativen Eigenschaften erstellt werden. Für die Umgebung kann oft eine modellorientierte Beschreibung (z. B. statecharts) erfolgen, die auch zur Wiederverwendung von Varianten der Maschine benutzt werden kann, während für die Maschine oft eine kombinierte Beschreibung von Vorteil ist. Sie erlaubt es insbesondere, die Eigenschaften direkt in der geeigneteren Form zu verwenden.

Die wesentliche Idee dabei ist, die Läufe eines Modells durch die Hinzunahme von TLA Formeln an den betreffenden Zuständen einzuschränken. Diese Vorgehensweise birgt natürlich die Gefahr in sich, die Läufe unangemessen einzuschränken. Somit tritt die Frage der Konsistenz einer solchen Beschreibung auf, d. h. die Frage, ob überhaupt noch Läufe möglich sind.

Es stellt sich heraus, dass für die hier betrachtete Erweiterung von statecharts mit TLA Formeln als Bedingung an die Zustände die Konsistenz i. Allg. unentscheidbar ist. Gewisse syntaktische Einschränkungen sowie eine geeignete Wahl der Zustände, an denen die Einschränkungen angeheftet werden, garantieren aber Konsistenz.

Wie Konsistenz und andere Eigenschaften einer Beschreibung lokalisiert und nachgewiesen werden können, wird ausführlich in [5] diskutiert. Eine Werkzeugunterstützung erweist sich als notwendig.

Diese Ergebnisse untermauern die Sinnhaftigkeit des unter 3.1 aufgestellten Modells für Problemspezifikationen, da wesentliche Instanzen davon in den Fallbeispielen vorkommen. Es unterstützt weiterhin die Verfolgbarkeit von Eigenschaften, da diese oft explizit an den wesentlichen Stellen der Modelle vorkommen, und eröffnet auch den Übergang einer Problem- zu einer Entwurfsspezifikation durch Verwendung von Verfeinerungsprinzipien.

Bewertung von Beschreibungstechniken. Der in [7] eingeführte Rahmen für die Analyse zeitbehafteter formaler Beschreibungstechniken basiert auf dem Institutionsbegriff von Goguen und Burstall. Dieser Rahmen legt die minimalen Eigenschaften auf syntaktischer und semantischer Ebene fest, die eine Beschreibungstechnik erfüllen muss, um zu diesem Typ von Beschreibungstechniken zu gehören. Typische Beschreibungstechniken, denen die Attribute formal und zeitbehaftet zugeordnet werden, sind die verschiedenen Varianten von Temporallogiken. In [8] wurde für die recht unterschiedlichen Temporallogiken MTL, CTL* und MTL-$\int$ explizit nachgewiesen, dass sie zu dem Typ formaler, zeitbehafteter Beschreibungstechniken gehören.
Der eingeführte Rahmen kann nun als Ausgangspunkt verwendet werden, um formale und präzise Kriterien für diesen Typ von Beschreibungstechniken aufzustellen, so dass ein objektiver Vergleich von Beschreibungstechniken möglich ist.

Vergleiche mit anderen Arbeiten außerhalb des SFB:

Kombination von Beschreibungstechniken. Es gibt eine Vielzahl von Untersuchungen zur Erweiterung und Kombination von formalen Beschreibungstechniken. Kombinationen von statecharts und TLA mit anderen Beschreibungstechniken wurden mehrfach untersucht, z. B. wird in [33], [34] die formale Semantik von statecharts mit anderen Methoden präzisiert. Eine Kombination von statecharts und Z wird in [35] verwendet, um das Verhalten reaktiver Systeme zu beschreiben. Zwei unterschiedliche Beschreibungen, nämlich zum einen eine Beschreibung des Lebenszyklus der Objekte in statecharts und zum anderen eine Beschreibung der Datenstrukturen und Invarianten in Z, werden benötigt. Der Zusammenhang dieser Beschreibungen (Konsistenzbedingungen) werden in Z beschrieben. Ebenfalls existieren einige Ansätze zur Kombination synchroner Beschreibungssprachen, etwa die Kombination von Argos und Lustre [36].

Bewertung von Beschreibungstechniken. In der Literatur sind weitere Sammlungen von Kriterien zur Bewertung von Beschreibungstechniken vorhanden, exemplarisch sei hier auf den Überblick [31] verwiesen. Bezüglich der immer mehr an Bedeutung gewinnenden Architekturbeschreibungssprachen verweisen wir auf den Überblicksartikel [32].

Offene Fragen:

Kombination von Beschreibungstechniken. Die Frage nach der *Konsistenz* von mit TLA Formeln kombinierten statecharts ist von zentraler Bedeutung. Wie kann also garantiert bzw. überprüft werden, ob ein mit TLA Formeln assoziiertes statechart noch Läufe zulässt?

Die Methode der schrittweisen *Verfeinerung* wird innerhalb der Entwicklung von Beschreibungen (Anforderungsspezifikation, Entwurfsspezifikation, usw.) oft eingesetzt. Damit ergeben sich u. a. folgende Fragen: Wie können die für TLA zur Verfügung stehenden Verfeinerungskonzepte auf die Kombination von statecharts mit TLA übertragen werden? Wie kann ein Verfeinerungsschritt verifiziert werden, und welche Möglichkeiten der Werkzeugunterstützung sind hierfür möglich?

Je größer und umfangreicher Beschreibungen werden, desto mehr werden Methoden zur *Strukturierung* dieser Beschreibungen erforderlich. Eine Möglichkeit der Strukturierung scheint durch activity-charts gegeben zu sein. Man kann in diesem Zusammenhang etwa folgende Fragen stellen: Wie können activity-charts zur Strukturierung des Verhaltens großer Systeme eingesetzt werden? Kann die in [17] beschriebene semantische Integration von activity-charts in der Kombination von statecharts und TLA die oben genannte Strukturierung unterstützen?

Bewertung von Beschreibungstechniken. Bisher wurde die Eignung des Rahmens nur durch die Einbettung einiger Beschreibungstechniken nachgewiesen. Es ist zu klären, ob der Rahmen auch geeignet ist, entsprechende Kriterien formal zu definieren. Unbeantwortet ist bisher auch die Frage, wie methodische Aspekte in diesen Rahmen integriert werden können.

4 Zusammenarbeit mit anderen Teilprojekten

Die aus der ersten Förderperiode bereits bestehende enge Kooperation mit Teilprojekt B4 wurde in der zweiten Förderperiode noch intensiviert und ist getrennt im Ergebnisbericht zu Querschnittsprojekt Q4 dokumentiert.

Darüber hinaus war das Teilprojekt auch an den Arbeiten aller anderen Querschnittsprojekte stark beteiligt. Im Rahmen von Querschnittsprojekt Q1 beteiligte sich ein Mitarbeiter aus Teilprojekt C1 als Entwickler und führte eine Vielzahl von Diskussionen mit Mitarbeitern aus Teilprojekt D1 über deren Prototyping-Ansatz im Vergleich zu dem in Querschnittsprojekt Q4 verwendeten Spezifikationsansatz.

Für das Querschnittsprojekt Q2 wurde in Zusammenarbeit mit Mitarbeitern der Teilprojekte A1 und B1 vorbereitend eine formale Problemspezifikation erstellt, die den Studenten als Grundlage bereitgestellt wurde. Des Weiteren stand ein Mitarbeiter aus Teilprojekt C1 als Ansprechpartner bei Rückfragen der Studenten zur Verfügung.

Durch die intensive Mitarbeit des Teilprojektes C1 in Querschnittsprojekt Q3 entstanden enge Verbindungen zu den Teilprojekten B10, B11, D1 und D2, die in der nächsten Förderperiode noch weiter vertieft werden sollen.

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] A. Schowalter. Modellierung einer Produktionszelle in einer Kombination aus statecharts und TLA. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1997.

[2] M. Kaiser: *Dokumentation der Problemspezifikation der Streckenbeeinflussungsanlage A 61 Bingen-Rheinböllen*. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999.

[3] F. Annawald: *Dokumentation der Problemspezifikation der Verkehrsbeeinflussungsanlage A 8 Hohenstadt-Riedheim*. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2000.

Diplomarbeiten

[4] T. Schmidt-Samoa: *FOREST - Entwurf und Implementierung einer Umgebung zur Erstellung formaler System-Anforderungs-Beschreibungen*. Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999.

Dissertationen

[5] T. Deiß: *On the Combination of Model- and Property-Oriented Description Techniques*. Dissertation, Fachbereich Informatik, Universität Kaiserslautern, 1999.

[6] M. Kronenburg: *Ein Ansatz zur Erstellung präziser, verständlicher Problemspezifikationen großer reaktiver Systeme*. Dissertation, in Vorbereitung, 2000.

SFB-Berichte

[7] R. Eschbach, T. Deiß, M. Kronenburg: *A Framework for the Analysis of Formal Description Techniques for Timed Systems*. Technischer Bericht SFB 501 05/98, Fachbereich Informatik, Universität Kaiserslautern, 1998.

[8] M. Kronenburg, T. Deiß, R. Eschbach: T*emporal Logics as Examples of Formal Description Techniques for Timed Systems*. Technischer Bericht SFB 501 07/98, Fachbereich Informatik, Universität Kaiserslautern, 1998.

[9] R. Gotzhein, M. Kronenburg, C. Peper: *Reuse in Requirements Engineering: Discovery and Application of a Real-Time Requirement Pattern.* Technischer Bericht SFB 501 08/98, Fachbereich Informatik, Universität Kaiserslautern, 1998.

[10] T. Deiß: *Case Study: Implementing an SDL System in Erlang*. Technischer Bericht SFB 501 06/99, Fachbereich Informatik, Universität Kaiserslautern, 1999.

[11] L. Geyer, M. Kronenburg, R. Merz, B. Schürmann: *Domain Modeling in the SFB 501*. Technischer Bericht SFB 501 07/00, Fachbereich Informatik, Universität Kaiserslautern, 2000.

Vorträge

[12] M. Kronenburg: *xforest: A Tool for the Creation of WWW-based Problem Specifications*. Dagstuhl-Seminar Requirements Capture, Documentation, and Validation, Dagstuhl, Deutschland, 1999.

Tagungsbeiträge

(Der Vortragende ist unterstrichen.)

[13] C. Peper, R. Gotzhein, M. Kronenburg: *A Generic Approach to the Formal Specification of Requirements*. Proc. 1st IEEE International Conference on Formal Engineering Methods (ICFEM‘97), Hiroshima, Japan, S. 252-261, 1997.

[14] R. Gotzhein, M. Kronenburg, C. Peper: *Reuse in Requirements Engineering: Discovery and Application of a Real-Time Requirements Pattern.* 5th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT‘98), LNCS 1486, Springer, Lyngby, Dänemark, S. 65-74, 1998.

[15] M. Kronenburg, C. Peper: *Definition and Instantiation of a Reference Model for Problem Specifications*. Proc. 11th International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Deutschland, S. 332-336, 1999.

[16] R. Gotzhein, M. Kronenburg, C. Peper: *Pattern-based Requirements Capture Applied: The SFB 501 Case Study.* In E. Börger, B. Hörger, D. Parnas, D. Rombach, Requirements Capture, Documentation, and Validation, Dagstuhl-Seminar-Report 242, 1999.

[17] T. Deiß: *An Approach to the Combination of Formal Description Techniques: Statecharts and TLA*. In K. Araki, A. Galloway, K. Taguchi, Proc. of 1st International Conference on Integrated Formal Methods, IFM‘99, S. 231-250. Springer, 1999.

[18] R. Eschbach: A *Termination Detection Algorithm: Specification and Verification*. In Jeanette M. Wing, J. Woodcock, J. Davies, Proc. of FM'99 - World Congress on Formal Methods in the Development of Computing Systems, LNCS 1709, Springer, Toulouse, Frankreich, S. 1720-1737, 1999.

[19] R. Eschbach, U. Glässer, R. Gotzhein, A. Prinz: *On the Formal Semantics of SDL-2000: A Compilation Approach using Abstract State Machines*. Erscheint in International Workshop on Abstract State Machines ASM'2000, LNCS, Springer, Monte Verità, Schweiz, 2000.

Zeitschriften / Monografien

[20] M. Kronenburg, C. Peper: *Application of the FOREST Approach to the Light Control Case Study.* Erscheint in Special Issue on Requirements Engineering: The Light Control Case Study, Journal of Universal Computer Science (J.UCS), Springer, *2000.*

6 Fremdveröffentlichungen

[21] P. Zave, M. Jackson: *Four Dark Corners of Requirements Engineering.* IEEE Transactions on Software Engineering and Methodology, 6(1), S. 1-30, 1997.

[22] M. Jackson: *A Discipline of Description (Keynote Talk)*. Requirements Engineering 3(2), S. 73-78, 1998.

[23] D. L. Parnas, J. Madey: *Functional documents for computer systems.* Science of Computer Programming, 25, S. 41-61, 1995.

[24] G. Booch, J. Rumbaugh, I. Jacobson: *The Unified Modeling Language User Guide*. Addison Wesley, 1999.

[25] M. Heisel: *Softwaretechnik und Methodik*. In Antrittsvorlesungen der Fakultät für Informatik. Universitätsschriften der Universität Magdeburg, 1998.

[26] H. Kilov, B. Rumpe: *Second ECOOP Workshop on Precise Behavioral Semantics (with an Emphasis on OO Business Specifications).* In S. Demeyer, J. Bosch, Object-Oriented Technology: ECOOP`98 Workshop Reader, S. 167-188, Springer, 1999.

[27] J. J. Odell, G. Ramackers: *Toward a Formalization of OO Analysis*. In J. J. Odell, Advanced Object-Oriented Analysis & Design Using UML, S. 43-55, Cambridge University Press, 1998.

[28] H.-D. Ehrich: *Object Specification*. In E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner, Algebraic Foundations of Systems Specification, S. 435-465, Springer, 1998.

[29] K. Lano: *Z++, An Object-Orientated Extension of Z.* In J. E. Nicholls, Z User Workshop, Oxford 1990, S. 151-172, 1991.

[30] J.C. Bicarregui, K.C. Lano, T.S.E. Maibaum: *Formalising Object-Oriented Models in the Object Calculus*. In J. Bosch, S. Mitchell, Object-Oriented Technology: ECOOP'97 Workshop Reader, S. 155-160, 1998.

[31] R. Wieringa: *A Survey of Structured and Object-Oriented Software Specification Methods and Techniques.* ACM Computing Surveys, 30(4), 1998.

[32] N. Medvidovic, R. N. Taylor: *A Framework for Classifying and Comparing Architecture Description Languages*. In M. Jazayeri, H. Schauer, Software Engineering - ESEC/FSE '97, S. 60-76, Springer, 1997.

[33] W. Damm, B. Josko, H. Hungar, A. Pnueli: *A compositional real-time semantics of STATEMATE designs.* In W.-P. de Roever, H. Langmaack, A. Pnueli, Proc. International Symposium Compos'97, Compositionality - The Significant Difference, LNCS 1536, Springer, S. 186-238, 1998.

[34] N. Day: *A Model Checker for Statecharts (Linking CASE Tools with Formal Methods).* Technical Report 93-35, Department of Computer Science, Vancouver, B.C. Canada, 1993.

[35] M. Weber: *Combining statecharts and Z for the design of safety-critical control systems*. In M. C. Gaudel and J. Woodcock, Proc. Intl. Symp. of Formal Methods Europe FME'96, LNCS 1051, S. 307-326, Springer, 1996.

[36] F. Maraninchi, Yann Rémond: *Mode-Automata: About Modes and States for Reactive Systems*. In C. Hankin, Proc. European Symposium on Programming, ESOP'98, LNCS 1381, Springer, S. 185-199, 1998.

Teilprojekt D1

Anwendungssystem Gebäude

Leiter

Prof. Dr. Gerhard Zimmermann	Dr. Bernd Schürmann
geb. 18.5.1939	geb. 22.8.1959
Universität Kaiserslautern	Universität Kaiserslautern
Postfach 3049	Postfach 3049
67653 Kaiserslautern	67653 Kaiserslautern
Tel.: 0631/205-2628	Tel.: 0631/205-3069
Fax: 0631/205-2162	Fax: 0631/205-2162
zimmerma@informatik.uni-kl.de	schuerma@informatik.uni-kl.de

Mitarbeiter

Ergänzungsausstattung:

Dipl. Inform. S. Queins
Dipl. Inform. J. P. Riegel (seit 1.8.1999)
Dipl. Inform. M. Schütze (bis 30.4.1999)

Grundausstattung:

Dipl. Inform. J. P. Riegel (bis 1.8.1999)
Dipl. Inform. A. Metzger (seit 1.12.1998)
Dipl. Inform. T. Tetteroo (seit 1.8.1999)

1 Kurzfassung der Aufgabenstellung

Primär dient dieses Teilprojekt als Infrastrukturprojekt für die B-Projekte. Es spielt die Rollen des Auftraggebers und, zusammen mit D2, des Experten im Anwendungsgebiet „Gebäudeautomation“. Desweiteren stellt D1 dem SFB Testumgebungen zur Verfügung. Daraus ergeben sich vier Arbeitsschwerpunkte:

Die Rolle des Experten wird durch die **Anwendungsfeldanalyse** erarbeitet. Dazu ist Wissen über das gesteuerte System, also die Physik der Gebäude und die Funktion der Installationen, und Wissen über Steuerungen notwendig. Unser Ziel ist es, das Wissen so zu strukturieren und als „Lexikon“ abzulegen, dass es automatisch oder interaktiv in allen Phasen des Software-Entwicklungsprozesses eingesetzt werden kann, ohne auf Anwendungsexperten zurückgreifen zu müssen.

In der Rolle des Auftraggebers werden **Szenarien und Problembeschreibungen** erstellt und gesammelt, die den Anwendungsbereich möglichst weit abdecken und vor allem auch nichtfunktionale Anforderungen enthalten.

Für **Prototyping und Test** von Steuerungssystemen werden **Gebäudesimulatoren** je nach Problembeschreibung in vielen Varianten benötigt. Deshalb wird ein System von Generatoren entwickelt, das für die statischen Komponenten die bereits vorhandenen Generatoren und für die dynamischen Komponenten generative Entwurfsmuster (ähnlich Design-Patterns) und neue Generatoren verwendet, um aus Problembeschreibung und Lexikon automatisch die richtige Variante zu erzeugen.

Simulatoren abstrahieren von der Realität. Ein endgültiger Test der Steuerungssysteme ist deshalb nur an einem realen Gebäude möglich. Außerdem müssen die Simulatoren getestet werden. Deshalb werden Teile von Gebäuden als **Testfeld** ausgebaut, das mit Sensoren, Aktuatoren, Prozessoren und Kommunikationstechnik ausgestattet wird.

Als zusätzliche Aufgabe wurde das ursprünglich in B9 beantragte **Requirements-Engineering** mit übernommen, da diese Phase im Entwicklungsprozess gerade bei komplexen eingebetteten Systemen besonders wichtig ist und effizient bearbeitet werden muss.

2 Entwicklung des Teilprojekts in der Förderperiode

Bei der Bewilligung des Teilprojektes D1 für den Berichtszeitraum wurde zur Auflage gemacht, Teile des nicht bewilligten Teilprojektes B9 mit zu übernehmen. Daraus ergab sich außer den in D1 geplanten Arbeitsschwerpunkten A bis D ein weiterer, den wir hier Arbeitsschwerpunkt E: Requirements-Engineering nennen. Zu Beginn der Periode gab es im SFB keine Gruppe, die sich mit dieser wichtigen Phase des Entwicklungsprozesses beschäftigt hat. Für Experimente mit späteren Phasen war es aber unbedingt notwendig, hier entsprechende domänenspezifische Lösungen zu finden und in Fallstudien Beispiele von System-Requirements bereitzustellen.

Dieser zusätzliche Arbeitsschwerpunkt hat sich als sehr fruchtbar herausgestellt. Allerdings mussten die Arbeiten in den Arbeitsschwerpunkten A und B entsprechend reduziert werden, was durchaus gerechtfertigt erscheint. Die Arbeitsschwerpunkte C und D wurden wie geplant durchgeführt.

Personell gab es Änderungen durch die Promotion von Herrn Schütze, auf dessen Stelle Herr Riegel aus der Grundausstattung nachrückte. In der Grundausstattung waren nicht immer beide Stellen besetzt, bedingt durch den allgemeinen Mangel an geeigneten Kandidaten in den schwachen Jahrgängen.

3 Durchgeführte Arbeiten und Ergebnisse

3.1 Arbeitsschwerpunkt A: Anwendungsfeldanalyse

Ein *Global-Dictionary* genanntes Hypertext-Dokument wurde kontinuierlich mit Wissen aus der Anwendungsdomäne gefüllt. Es besteht zur Zeit aus etwa 300 Objekten, die in 6 Kategorien geordnet und zusätzlich über Generalisierungs-, Aggregations-, Material- und Topologie-Relationen verknüpft sind. Navigiert wird über Hyperlinks. Auf diese Weise sind die professionellen englischen Bezeichnungen mit Erläuterungen, physikalischen Zusammenhängen und Standardwerten leicht aufzufinden, wenn ein Schlagwort in der logischen Nähe des Objektes bekannt ist. Da das Global-Dictionary Wissen über die gesteuerte Strecke mit Wissen über Steuerung und Automation verknüpft, ist seine Struktur ideal geeignet, um Artefakte für die Wiederverwendung abzulegen und zu finden. Benutzt wurde das Global Dictionary vor allem zur Erstellung von Auszügen für spezielle Projekte, den *Project-Dictionaries.*

3.2 Arbeitsschwerpunkt B: Szenarien und Problembeschreibungen

In einer umfangreichen Studie wurde als Szenario eine existierende Heizungs-, Klima- und Lichtsteuerung in einem Großraumbüro (Intelligent Workplace, CMU) mit individueller Arbeitsplatzsteuerung analysiert und modelliert. Die untersuchten Steuerungen waren das Metasys-System von Johnson Controls für Klima und Heizung und EIB von Siemens für Licht. Die Klimatisierung der einzelnen Arbeitsplätze wurde durch Personal Environment Module von Johnson Controls ermöglicht. Zur Beeinflussung des natürlichen Lichteinfalls wurden drehbare Spiegel und Jalousetten gesteuert. Speziell zur Modellierung wurden eine Reihe von Ansätzen erprobt, die schließlich zu der im Arbeitsschwerpunkt E beschriebenen Analysemethode auf der Basis einer angepassten Modellarchitektur geführt haben [13], [21].

Da die analysierte Steuerung für den futuristischen Intelligent Workplace (IW) sehr konservativ ist, wurde zusammen mit Prof. Mahdavi (CMU) als gemeinsames Projekt eine Dissertation an der CMU initiiert, die den technischen Möglichkeiten des IW entsprechende Automationsszenarien entwickeln soll.

Als Infrastrukturprojekt wurden Problembeschreibungen für Fallstudien erstellt und es wurde in der Rolle des Kunden und des Domänenexperten (zusammen mit D2) Hilfestellung gegeben. Die Problemstellung für eine Lichtsteuerung wurde ein Art Benchmark für einen internationalen Vergleich verschiedener Methoden zum Requirements-Engineering (siehe auch [28]). Eine Erweiterung dieser Problembeschreibung zur Temperatursteuerung wurde Grundlage der im Querschnittsprojekt Q3 durchgeführten Domänenanalyse [19].

3.3 Arbeitsschwerpunkt C: Gebäudesimulation

Ziel dieses Arbeitsschwerpunktes ist es, eine Umgebung bereitzustellen mit der es einfach und schnell möglich ist, angepasste Gebäudesimulatoren zu erzeugen. Als Modellierungsnotation werden dazu neben Klassendiagrammen vor allem domänenspezifische Entwurfsmuster (Pattern) verwendet. Bereits in der ersten Förderungsperiode entstanden die Grundideen und erste Werkzeuge. Die folgende Aufzählung zeigt die wesentlichen Neuerungen dieser Periode:

- Implementierung weiterer Werkzeuge.
- Strukturierung des Pattern-Kataloges. Neu hinzugekommen sind vor allem die Abschnitte Software-Architektur und Kontrollfluss.
- Typisierung der Pattern-Schnittstellen und Aufstellung von Regeln zur Anwendung der Entwurfsmuster.
- Erprobung der Entwurfsmethode und Überprüfung der Simulatoren.

Im Rahmen von AP-C1 wurden neue Entwurfsmuster entwickelt und Regeln und Typen für bestehende Pattern eingeführt. AP-C2 beschäftigte sich vor allem mit der Integration verschiedener Komponenten und Modelle. Die Erprobung der Entwurfsmethode für Gebäudesimulatoren fand im Rahmen von AP-C3 statt.

Arbeitspaket AP-C1: Modelle

Der aktuelle Pattern-Katalog [29] besteht aus 41 Pattern aus den Bereichen Software-Architektur, Kontrollfluss, Benutzungsschnittstelle und Gebäudesimulation. Etwa die Hälfte der Entwurfsmuster wurde während dieser Förderungsperiode entworfen. Diese Muster können zur Modellierung von Software-Komponenten verwendet werden. Dazu werden sie instanziiert und an das zu entwerfende Komponentenmodell formal angebunden [26]. So dienen die Pattern zum einen zur Festlegung der Funktionalität eines oder mehrerer Objekte und sie können darüber hinaus als Bindeglied zwischen verschiedenen Modellen und Notationen eingesetzt werden (siehe auch AP-C2). Generatoren überführen die Modelle in fertig ablauffähigen Programmcode [24], [25]. Die entwurfsmusterbasierte Modellierung von Komponenten in der Domäne Gebäudesimulation wurde von Martin Schütze in seiner Dissertation [12] zusammengefasst und erläutert.

Um den Umgang mit den Pattern zu vereinfachen und den Entwurfsspielraum sinnvoll einzuschränken, wurde eine Typisierung der Pattern-Schnittstelle eingeführt [6]. Dadurch wird die Anbindung der Pattern einfacher, da unsinnige Bindungspunkte und falsche Pattern-Kombinationen von vorn herein verboten sind. Neben der Typisierung der anbindbaren Methoden und Attribute können mögliche Bindungspunkte auch durch Regeln weiter eingeschränkt werden. Damit ist es möglich, bestimmte Objektstrukturen zu fordern, um ein Pattern anbinden zu können. Innerhalb dieser Förderungsperiode entstand ein Editor zur Anwendung der Pattern [1]. Dieser Editor wurde zudem um das Konzept der Typen und Regeln erweitert, sodass nur gültige Pattern-Bindungen erstellt werden können [6].

Entwurfsmuster können weiterhin zur Modellierung der Architektur einer Software-Komponente benutzt werden. Einige Architekturstile (z.B. Pipes&Filters, Model-View-Controller) können durch Pattern ausgedrückt werden und so explizit in die Modellierung einfließen [20]. Somit wird eine schrittweise Verfeinerung der Komponentenmodelle unterstützt, indem zunächst Architekturmerkmale modelliert werden und diese dann (mit demselben Beschreibungsmittel) später konkretisiert werden.

Arbeitspaket AP-C2: Synthese

Ein wichtiger Aspekt in der Modellierung von Gebäudesimulatoren ist die klare Definition und Verwendung von Schnittstellen zwischen einzelnen Teilen. Dieses sind sowohl Schnittstellen innerhalb eines Modells (z.B. Verteilung, Kopplung unterschiedlicher Effekte), als auch externe Schnittstellen (Kommunikation mit der Hardware, Anbindung weiterer Modellnotationen). Zur Schnittstellendefinition zwischen Objekten dienen Pattern der Kategorie Software-Architektur. Diese helfen zum einen, unabhängig von der Objektinstanziierung zu werden (Patterns Traversal und Follow-Relation [29]) und zum anderen, Datenkanäle und Datenfluss zu modellieren (Pattern Pipes&Filters [7]). Die Pipes&Filters Pattern bieten darüberhinaus auch eine einfache Möglichkeit zur Verteilung eines Simulators auf mehrere Rechner. Die dabei auftretende Kommunikation geschieht mit dem in [30] beschriebenen socket-basierten Mechanismus.

Im Rahmen einer Diplomarbeit [8] wurde die Integration Endlicher Automaten in Klassendiagramme mit Pattern untersucht. Dabei ging es darum, Kontrollfluss und Funktionalität einzelner Objekttypen mit Endlichen Automaten (z.B. in SDL) zu beschreiben und diese dann an ein Klassendiagramm zu binden. Die dabei entstandenen Pattern können benutzt werden, um verschiedene Abarbeitungsstrategien der Endlichen Automaten auszuwählen.

Gegen Ende dieser Antragsperiode wurden Untersuchungen angefangen, welche Auswirkungen die Kopplung eines Simulators mit realer Hardware (Testfeld) hat. Damit bei gemischter HW/SW-Simulation möglichst wenige gegenseitige Beeinflussungen auftreten, ist eine gute Partitionierung zwischen Simulator und Gebäudeinstallation nötig. Diese hängt im wesentlichen vom betrachteten Szenario und den zu messenden Größen ab.

Arbeitspaket AP-C3: Erprobung

In der Antragsperiode wurden zahlreiche Fallstudien durchgeführt, um die beschriebene entwurfsmusterbasierte Modellierungsmethode zu verbessern und um die mathematische Genauigkeit der Simulatoren zu überprüfen. Die Modellierungsexperimente dienten in erster Linie zur Verbesserung des Pattern-Kataloges, indem neue Pattern definiert wurden und bestehende Pattern verfeinert und korrigiert wurden. Die Parameter der einzelnen Simulationsobjekte wurden durch Vergleichsmessungen mit dem realen Testraum abgestimmt. Zum Vergleich des entwurfsmusterbasierten Ansatzes mit der SDL Modellierungsmethode wurde ein Simulator zur Wärmesimulation in SDL modelliert (siehe auch 3.5). Diese Arbeiten sind noch nicht abgeschlossen, zeigen aber bereits, dass der von SDT automatisch generierte C-Code die thermische Simulation von Gebäuden mit etwa 1000 Räumen in Realzeit zulässt.

In weiteren Untersuchungen wurde festgestellt, dass die Geschwindigkeit der generierten Simulatoren im Wesentlichen von der Zahl der vorkommenden Prozesswechsel abhängt. Die Zeit zur eigentlichen Berechnung der Simulationsformeln ist relativ klein gegenüber der Event-Verarbeitung und den damit verbundenen Prozesswechseln. Ein Optimierungsansatz ist es also, die Prozesswechsel beim Simulator möglichst gering zu halten. Dieses kann durch Delegation der Berechnungen an aggregierte Objekte geschehen. Dabei dürfen jedoch keine Zeitgrenzen verletzt werden, die die Simulationsergebnisse (durch zu seltene Berechnungen) falsch werden lassen. Die Anzahl der in Realzeit simulierbaren Räume (mit 6 Prozessen pro Raum) liegt bei ca. 200. Wird das gesamte Haus nur als ein Prozess gerechnet, lassen sich ca. 1000 Räume in Realzeit simulieren.

3.4 Arbeitsschwerpunkt D: Testfeld

Das Testfeld wurde um drei weitere Räume und zwei Flursegmente erweitert, womit ein größerer Bereich von Szenarien zur Validierung von Steuerungssystem-Prototypen und Simulatoren abgedeckt wird (siehe AP-D1). Bei der Installation zusätzlicher Komponenten wurde die bisher eingesetzte sternförmige Verdrahtung aufgegeben und durch eine Kommunikationshierarchie ersetzt. Dabei sind die einzelnen Ebenen in dieser Hierarchie über unterschiedlich leistungsfähige und skalierbare Kontroller (Prozessoren) gekoppelt (siehe AP-D2). Dieses Testfeld mit ca. 115 Sensoren und 55 Aktuatoren wurde zur Durchführung von Experimenten genutzt, bei denen D1 Hilfestellung leistete (siehe AP-D3).

Arbeitspaket AP-D1: Sensoren und Aktuatoren

Es hat sich herausgestellt, dass zur Beurteilung unterschiedlicher Steuerungstrategien für Gebäudeautomationssysteme Experimente parallel zueinander durchgeführt werden müssen. Externe Einflüsse wie z.B. das Wetter sind nicht beeinflussbar und deshalb sind Vergleichsexperimente, die zu unterschiedlichen Zeitpunkten durchgeführt werden, schwieriger zu beurteilen. Daher wurde die initiale Installation [13], die aus einem Computerlabor bestand, das von Studenten genutzt wurde, zunächst um einen weiteren solchen Raum ergänzt. Anschließend wurden zwei Mitarbeiterbüros, die sich nicht im Aufbau aber in der Nutzung von den Laboren unterscheiden, zum Testfeld hinzugenommen.

In jedem der Räume stehen Helligkeits- und Temperatursensoren, Bewegungsmelder und Fenster- und Türkontakte für Messungen und Eingaben in das Steuerungssystem zur Verfügung. Die Heizkörperventile können durch elektrische Stellantriebe gesteuert werden. Im Notlaufbetrieb sorgt ein lokaler PD-Regler für deren Ansteuerung. Die Raumbeleuchtung wurde so modifiziert, dass sie sich elektronisch schalten und dimmen lässt. Das Schalten der Lampen bleibt durch den Einsatz von Stromstoßrelais unabhängig vom Automationssystem. Zur Abdunklung der Räume wurden diese mit Innenjalousien ausgestattet, deren Lamellen sich motorisch bewegen lassen. Für einen typischen Raum kann man so auf 22 Sensoren und 12 Aktuatoren zugreifen. [30]

In den zwei Flursegmenten wurden 18 Bewegungsmelder montiert, welche Bewegungen im Flur und vor den Zugangstüren erfassen. Die Beleuchtung der Flursegmente kann über Pulsgeneratoren geschaltet werden. An den Außenwänden des Gebäudes wurden Helligkeitssensoren angebracht, mit deren Daten sich die in einen einzelnen Raum potenziell einfallende Helligkeit ermitteln lässt. Diesen wurden speziell auch für die Ausführung der in Querschnittsprojekt Q1 entstandenen Prototypen eingesetzt.

Der Einsatz von kostengünstigen Dallas 1-Wire-Sensoren wurde zu Gunsten einer Eigenentwicklung [2] aufgegeben, da wegen des aufwändigen Busprotokolls eine Ankopplung der Dallas-Bausteine über ein FPGA [3] nötig war.

Arbeitspaket AP-D2: Kontroller und Kommunikation

Es wurde die in Abbildung 1 gezeigte Kommunikationshierarchie realisiert. Auf unterster Ebene werden Sensoren und Aktuatoren über einfache Messdatenverbindungen an Mikrokontroller (μC) angekoppelt. Diese sind in einer Area (Raum oder Flursegment) untereinander und mit einem wesentlich leistungsfähigerem Embedded-PC über einen CAN-Bus vernetzt, um eine echtzeitfähige Kommunikation zu gewährleisten. Die Embedded-PCs aller Areas kommunizieren über Ethernet und besitzen eine Verbindung zu einem wiederum leistungsfähigeren Standard-PC. Das gesamte System lässt sich mit dem Uni-LAN und dem Internet verbinden, um auch von extern auf das Testfeld zugreifen zu können.

Für die Anbindung der Sensorik und Aktuatorik an die 8051-kompatiblen Mikrokontroller wurde ein modulares Design realisiert. Dabei werden die vom Mikrokontroller verfügbaren I/O-Pins so gruppiert, dass vorgefertigte I/O-Module angeschlossen werden können. Die Konfiguration erfolgt über den CAN-Bus ohne Hardwareeingriff und erlaubt so ein hohes Maß an Flexibilität für die eventuell in Experimenten notwendigen Umbauarbeiten und die daraus folgenden Änderungen in der Konfiguration. Neben der Umwandlung der Messwerte in physikalische Größen und der Kommunikation dieser Daten über den CAN-Bus laufen auf den Mikrokontrollern rudimentäre Steuerungsalgorithmen, die einen Minimalbetrieb der Installation gewährleisten [2]. Für Experimente während der Design-Phase bieten die Kontroller noch genügend Ressourcen für kleinere Komponenten des Steuerungssystems. Der Einsatz von SLIOs (Serial Linked I/O, [5]) zur Anbindung von Sensoren und Aktuatoren über CAN wurde wegen des eingeschränkten Adressbereichs nicht weitergeführt.

Die Installation wurde so verwirklicht, dass die Kopplung der einzelnen Areas auf drei unterschiedliche Arten erfolgen kann:

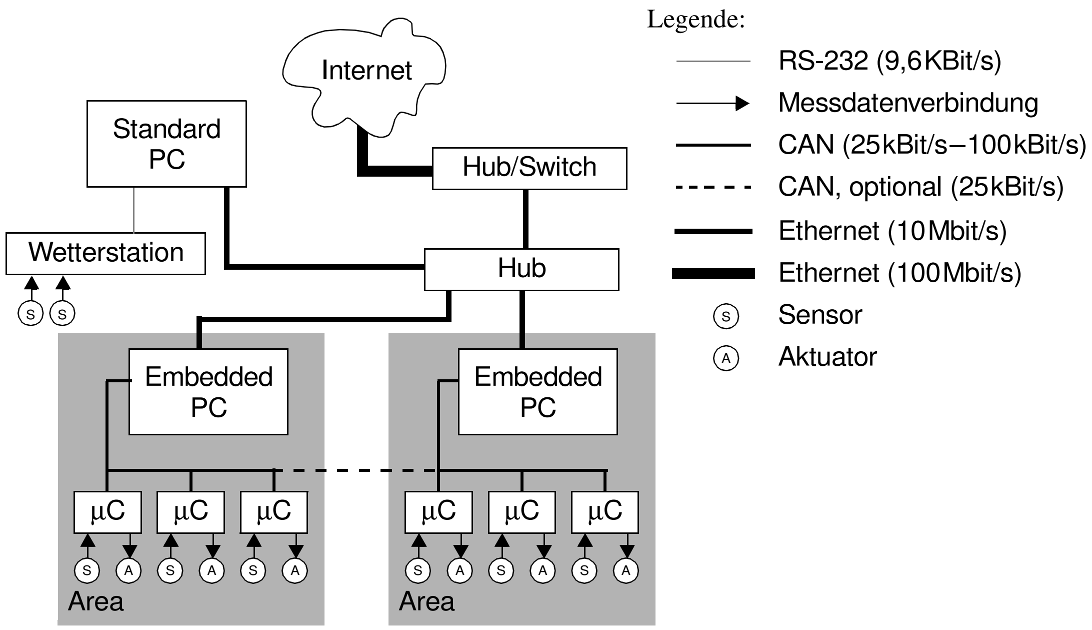

Abbildung 1: Kommunikationshierarchie

- Die einzelnen CAN-Bussegmente können zu einem großen Segment zusammengeschaltet werden, was die direkte Kommunikation aller Komponenten am Bus ermöglicht (in Abb. 1 durch die gestrichelte Verbindung zwischen den µC angedeutet).
- Die Embedded-PCs können eingesetzt werden, um CAN ↔ Ethernet-Router zu realisieren. Um das Routing und eine verbindungsorientierte Kommunikation der Komponenten zu ermöglichen, soll noch in dieser Förderungsperiode ein CAN-Protokoll aus Teilprojekt B4 eingesetzt werden. Dazu wurden Vorarbeiten für ein echtzeitfähiges Kommunikationssubsystem [10], [27] und einen CAN ↔ CAN-Router [11] geleistet.
- Der auf den Embedded-PCs vorhandene TCP/IP-Stack kann verwendet werden, um über eine Socket-Schnittstelle das für den Datenkonzentrator und den Simulator genutzte Protokoll [30] zur Verfügung zu stellen.

Auf den Embedded-PCs können in der Analyse oder Design entstandene Prototypen ausgeführt werden. In diesen Prototypen werden typischerweise komplexere Steuerungsaufgaben realisiert. Eine Kopplung zwischen einzelnen Prototyp-Komponenten auf unterschiedlichen Rechnern erfolgt über TCP/IP-Socket-Verbindungen.

Der Standard-PC wird zur Aufzeichnung von Messdaten und zum Anschluss der Wetterstation genutzt. Für die Administrierung des Systems existiert ein Java-Applet, das die Konfigurationsdaten sämtlicher Sensoren und Aktuatoren verwaltet.

Arbeitspaket AP-D3: Experimente

Zur Validierung des in Querschnittsprojekt Q1 spezifizierten Gebäudeautomationssystems wurde dieses um die nicht ausgestatten Räume reduziert und im Testfeld als Prototyp ausgeführt. Der Schritt von der Entwicklungsplattform (Sun-, bzw. HP-Workstation) zur Zielplattform (Embedded-PC) gestaltete sich im ersten Schritt sehr einfach. Es musste lediglich der erzeugte C-Code neu übersetzt und einige handgeschriebene C-Funktionen angepasst werden. Die Verteilung des Prototyps auf die Embedded-PCs war dagegen aufwändiger. Zunächst mussten sinnvolle Partitionen festgelegt und dann die Teilsysteme neu generiert werden, wobei für die Kopplung über Sockets, bzw. CAN, eine Kodierung und Dekodierung von Signalen erfolgen musste. Desweiteren wurde für die Socket-Kommunikation eine zentrale Komponente entwickelt, die alle Signale entgegennimmt und an das entsprechende Teilsystem weiterreicht. Es hat sich gezeigt, dass die einzelnen Schritte sehr mühsam sind und für effizientes Prototyping unbedingt durch Werkzeuge unterstützt werden müssen.

Als weiteres Experiment wurde ein Praktikum durchgeführt, in welchem Studenten ein Steuerungssystem modellierten. Nachdem das System mit Hilfe einer Simulation des Gebäudes getestet worden war, wurde ein realer Raum des Testfeldes zur Validierung genutzt.

3.5 Arbeitsschwerpunkt E: Requirements-Engineering

Wir haben an der Entwicklung einer Methode gearbeitet, die es erlaubt, aus einer gegebenen, informellen Problembeschreibung formale Anforderungen herzuleiten. Ziel des Ansatzes ist es, schnell ein ausführbares Modell zu erhalten, das dann, als Prototyp, sowohl zur Absprache mit dem Kunden, als auch zur Überprüfung der durchgeführten Entwicklungsschritte dient. Es dient weiterhin als Anforderungsdokument für die nachfolgenden Entwicklungsschritte. Bei der Entwicklung der Methode haben wir Wert auf die Effizienz der Entwicklung und eine verständliche und vollständige Formulierung der Anforderungen gelegt.

Die speziellen Eigenschaften der Methode sind in der folgenden Aufzählung genannt und werden im Weiteren detaillierter dargestellt. Die dabei referenzierten Fallstudien werden am Ende dieses Abschnitts kurz vorgestellt. Eine vollständige Beschreibung der Methode findet sich in [14], Kapitel 4.3.

- domänenspezifische Modellarchitektur
- spezielle informelle Beschreibung
- schrittweise Überführung und Verfeinerung in eine formale Beschreibung mit Inspektion und Prototyping
- Wiederverwendung von Domänenwissen und Entwicklungsprodukten
- Definition einer Metrik zur Bewertung der Wiederverwendung
- Unterstützung des Prototypings
- Analyse der Modellierung und Betrachtung der nichtfunktionalen Eigenschaften (speziell die Zeitanforderungen)
- Formalisierung des Ansatzes durch eine Entwurfssystematik

Die Entwicklungsschritte und -produkte sind in Abbildung 2 zusammengefasst.

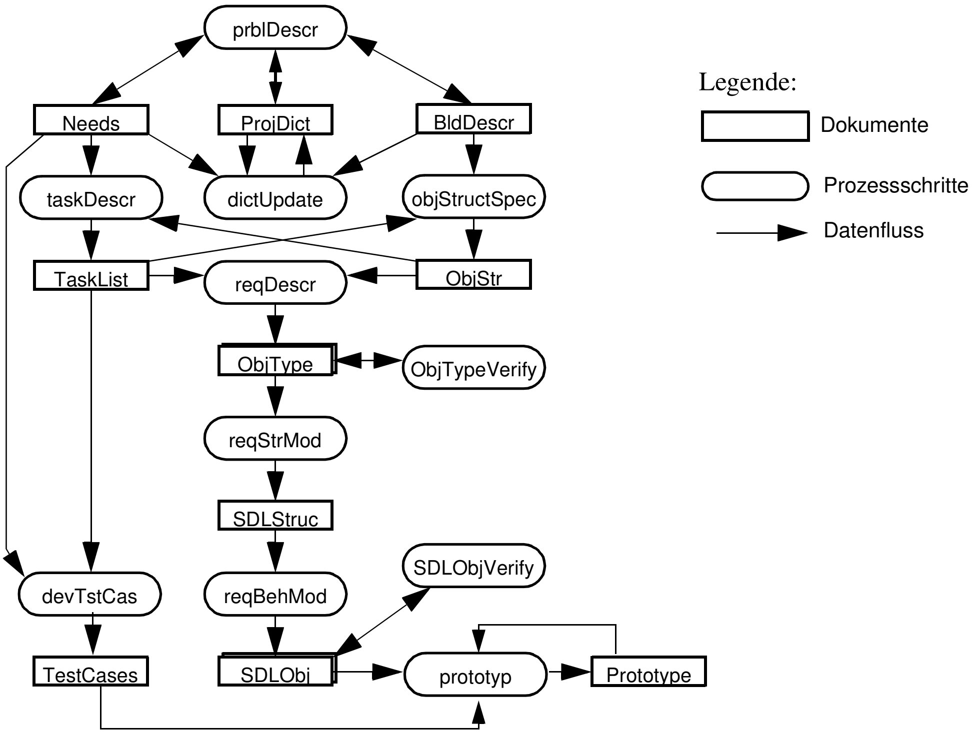

Abbildung 2: Prozessmodell

Zentraler Bestandteil ist die frühe Ermittlung einer Systemstruktur. Dies wird durch die Spezialisierung auf die Domäne Gebäudautomation ermöglicht, da sich eine sinnvolle Struktur für ein Steuerungssystem leicht aus der Struktur des Gebäudes ableiten lässt. Die Systemstruktur wird durch Aggregationsbeziehungen beschrieben, bei denen die einzelnen aggregierenden Typen nicht nur strukturierenden Charakter haben. Vielmehr werden ihnen, ähnlich einer Organisationshierarchie, Aufgaben (Tasks) zugewiesen [15]. Die Ermittlung der Tasks geschieht aus den in der Problembeschreibung gegebenen Anforderungen (Needs). Dabei kann ein Need durch mehrere Tasks bearbeitet werden, bzw. kann ein Task mehrere Needs realisieren. Die Tasks werden den einzelnen Objekttypen aus der Systemstruktur zugewiesen, woraus die Notwendigkeit der Zerlegung der Needs in einzelne Tasks folgt. Die Tasks werden informell beschrieben. Sie definieren, was ein Objekttyp im einzelnen zu leisten hat. Wie die einzelnen Tasks gelöst werden und welche Kommunikation sie dazu benötigen, wird als Strategie informell modelliert. Daraus ergeben sich auch die Schnittstellen zwischen den Objekttypen. Anschließend werden dann die Objekttypen in eine formale Beschreibung überführt. Zur Zeit haben wir als formale Beschreibungstechnik SDL gewählt, wobei unsere Methode nicht unbedingt auf diese Technik eingeschränkt ist.

Für die einzelnen Entwicklungsschritte sind Überprüfungen vorgesehen, die entweder durch eine Inspektion (speziell der informellen Dokumente) und durch Prototyping durchgeführt werden. Für die Inspektion wurden typabhängige Checklisten erstellt. Diese Checklisten wurden im Laufe der Entwicklung der Methode verfeinert bzw. erweitert. Als Beispiel sei die Fallstudie CS3/1 erwähnt ([16], Kapitel 5). Hier wurden die gefundenen Defekte in verschiedene Klassen eingeteilt. Die Konsequenz aus dieser Auswertung war die Verbesserung der Checklisten, um die häufig gemachten Fehler früher zu erkennen. Mit dieser Änderung und der Einführung eines Project-Dictionary konnten in der nächsten Fallstudie z.B. die Fehler bzgl. inkonsistenter Signaldefinition um ca. 70% reduziert werden.

In den einzelnen Fallstudien wurden neben der objekttypspezifischen Erfassung des Modellierungsaufwands die Anzahl und der Typ der Fehler erfasst. Daraus lassen sich dann die benötigten Aussagen bzgl. Fehlerausbreitung, Wiederverwendung und bei erneuten Projektdurchläufen bzgl Verbesserungen in der Methode ableiten.

Von der Wiederverwendung (Reuse) wird bei allen Entwicklungsschritten Gebrauch gemacht. Dazu werden die in früheren Projekten entwickelten Produkte in eine Reuse-Bibliothek eingebracht und so dem aktuellen Projekt zur Verfügung gestellt. Um den Nutzen einer speziellen Wiederverwendung beurteilen zu können, muss der Aufwand zur Neuerstellung eines Artefakts mit dem Aufwand zur Anpassung eines passenden Reuse-Artefakts verglichen werden. Um diesen abschätzen zu können, wird auf die Erfahrung aus früheren Wiederverwendungen zurückgegriffen. Desweiteren können wir am Ende eines Projekts Aussagen über den Nutzen von Wiederverwendung bezogen auf das gesamte Projekt machen. Dazu haben wir die in [32] vorgestellte Metrik auf unsere Modellierungsmethodik angepasst (siehe auch [14], Kapitel 7.2). Für die Fallstudie CS3/2 ergab sich eine Produktivitätssteigerung von 1,22 durch konsequente Wiederverwendung.

Ein zentrales Produkt unserer Methode ist das Project-Dictionary, in dem in höchstem Maße von Wiederverwendung Gebrauch gemacht wird. Es dient zur Kommunikation der Entwickler untereinander. In ihm werden Begriffe und objektübergreifende Informationen abgelegt. Initial wird es mit Daten aus dem projektübergreifenden Lexikon zur Verfügung gestellt und im Laufe der Entwicklung um die projektspezifischen Einträge erweitert. In der Fallstudie CS3/2 enthielt das Project-Dictionary am Ende des Projekts ca. 280 Einträge.

Die Erstellung eines Prototyps wird stark von den in der Entwicklungsumgebung SDT von Telelogic integrierten Generatoren unterstützt. Diese überführen ein SDL-Modell in ablauffähigen C-Code. Wie Abbildung 3 zeigt, stellen wir zur Anbindung der Prototypen an die Testumgebung (bei uns i. Allg. ein reales Gebäude oder eine Simulation des Gebäudes) die generische Komponente ProtoCtrl zur Verfügung. Diese ermöglicht die Verbindung entweder über eine TCP/IP-Socket-Schnittstelle oder, zur Kommunikation mit bereits installierten Hardware-Komponenten, über eine CAN-Schnittstelle. Der Transport der physikalischen Werte von bzw. zu den Sensor- und Aktuatorkomponenten innerhalb des Steuerungssystems erfolgt durch SDL Remote-Procedure-Calls. Die Kommunikation zur Umgebung wird durch die Java Komponente ControlSI [9] bzw. durch deren Erweiterung Interlink [31] ermöglicht.

Neben der Ermittlung der funktionalen Anforderungen wird Wert auf die nichtfunktionalen Anforderungen gelegt. In Bezug auf Zuverlässigkeit werden die Anforderungen betrachtet, die bereits vom Kunden definiert wurden. Diese lassen sich dann auch

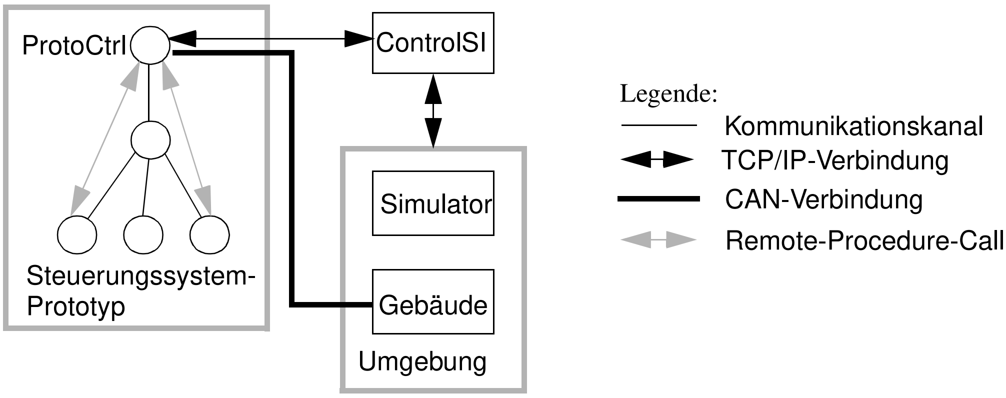

Abbildung 3: Kopplung eines Prototyp an die Umgebung

mit Hilfe des Prototypings überprüfen. Dies gilt nicht für die Zeitanforderungen, da bei unserem Ansatz für das Prototyping in dieser frühen Phase gerade bezüglich der Kommunikation sehr starke Vereinfachungen gemacht werden. Deswegen werden die Zeitanforderungen im SDL-Modell nicht explizit modelliert, sie werden vielmehr aus den SDL-Modellen hergeleitet. Dies betrifft die Zeitspanne, die nach Auslösen eines Event (z.B. die Aktion eines Benutzers) benötigt wird, bis eine Reaktion an einem Aktuator zu beobachten ist. Ausgehend von dem Event wird eine Kette von Signalen bzw. aufgerufenen Prozessen gebildet, die ausgehend von diesem Event bis hin zur Aktuatoränderung im System entsteht. Die Ermittlung dieser Kette geschieht mittels einer dynamischen Analyse [17], bei der sogenannte Trace-Files untersucht werden. Diese Trace-Files werden beim Ablauf des Prototyps erstellt und beinhalten die Kommunikation der Prozesse untereinander. Der Signalkette kann dann eine Zeit zugeordnet werden, innerhalb derer sie abgearbeitet werden muss.

Die dynamische Analyse dient neben der Ermittlung der Zeitanforderungen auch zur Bewertung der SDL-Modelle und wird auch im weiteren Entwurf, speziell der Designphase verwendet [17].

Um die Planung eines Projekts zu unterstützen, wurden verschiedene Abhängigkeiten zwischen den Objekten und ihrem Entwicklungsaufwand untersucht. Es hat sich herausgestellt, dass eine Abschätzung der Komplexität der Objekttypen stark mit dem benötigten Aufwand korreliert ([16], [14], Kapitel 7.1). Die Komplexität eines Objettyps errechnet sich aus der Summe der Komplexitäten seiner Tasks und der Anzahl seiner aggregierten Komponenten. Für die Fallstudie CS3/2 ergab sich eine Korrelation von Komplexität zu dem benötigten Modellierungsaufwand von 0,82 im Schnitt.

Zur Zeit wird an der Systematisierung unserer Methode gearbeitet. Wir haben begonnen, die Entwicklungsprodukte, Teilprozesse und andere benötigte Daten abstrakt zu beschreiben. Dazu haben wir zunächst ein die Produktdaten beschreibendes UML-Modell erstellt, von dem wir dann eine Abstraktion bilden. In diesem abstrakten Modell werden Versionierungsbeziehungen und Zusammenhänge zwischen Produkten, Teilprozessen und Werkzeugen beschrieben [18]. Das abstrakte Modell soll jedoch nicht nur zur Beschreibung der Methode des Requirements-Engineering dienen. Vielmehr soll die Möglichkeit bestehen, alle in der weiteren Entwicklung eingesetzten Prozesse in dieses Modell zu integrieren, um die Schnittstelle zwischen diesen Prozessen zu vereinfachen.

Zur Entwicklung und Erprobung unserer Methode wurde eine Reihe von Fallstudien durchgeführt. Diese sind chronologisch mit einer kurzen Beschreibung aufgelistet:

- **IntelligentWorkplace**: An der Carnegie Mellon University wurden Teile der Steuerung des dort entstehenden Intelligent Workplace modelliert (siehe Arbeitsschwerpunkt B).
- **CS3/1**: Für das Gebäude 32, 4. Stock wurde eine Lichtsteuerung im Rahmen eines Querschnittprojektes realisiert. Zum ersten Mal wurde ein größeres Projekt mit mehreren Personen durchgeführt.
- **CS3/2**: Die Problemstellung von CS3/1 wurde um eine Heizungssteuerung erweitert. Die Erfahrungen aus CS3/1 und die Einführung von Reuse wurden zur Prozessverbesserung eingesetzt. Die resultierenden Veränderungen am Prozess wurden anhand eines erneuten Projektdurchlaufes bewertet.
- **CSD1**: Es wurde die Problembeschreibung von CS3/1 noch einmal bearbeitet, jedoch wurde das in CS3/2 entwickelte System als Ausgangspunkt genommen und daraus die Temperatursteuerungskomponente entfernt. Gezeigt wurde dadurch die Anpassbarkeit der von uns entwickelten Produkte. Im Vergleich dazu wurde in **Light32** dieselbe Problemstellung ohne Wiederverwendung des gesamten CS3/2-Entwurfs bearbeitet. Ähnlich wie in CS3/2 wurde lediglich Wiederverwendung auf Objekttypebene betrieben [28].

Neben den aufgelisteten Fallstudien haben wir eine Reihe kleinerer Projekte durchgeführt. Wir haben einen alternativen Ansatz zur Modellierung der Systemstruktur erprobt [4] und unsere Methode bei der Modellierung und Generierung verschiedener Simulatoren eingesetzt. Gerade hier hat sich die Notwendigkeit gezeigt, neue Beschreibungstechniken wie Petri Netze oder Message-Sequence-Charts in die Methode zu integrieren. Desweiteren haben wir die Änderbarkeit unserer Entwürfe untersucht. Dazu haben wir bestehende Entwürfe bzgl. ihrer Funktionalität und ihrer räumlichen Struktur variiert.

Als Zusammenfassung wird in der folgenden Tabelle eine Überblick über vier Fallstudien gegeben (siehe auch [28]). Es werden die Komplexität des Gesamtprojekts (einheitenlos) und der Gesamtaufwand in Minuten angegeben. Die letzten beiden Zeilen geben den Aufwand pro kleinster Komplexitätseinheit und die Produktivität, normiert auf CS3/1, an.

Vergleich Komplexität und Aufwand	CS3/1	CS3/2	Light32	CSD1
Gesamtkomplexität	688	1826	1190	1190
Gesamtaufwand (min)	19000	29000	17100	1600
Aufwand / Komplexität (min)	27,6	15,9	14,4	1,34
Relative Produktivität	1	1,73	1,92	20,6

4 Zusammenarbeit mit anderen Teilprojekten

Eine enge Zusammenarbeit bestand durch die gemeinsame Arbeit in den Querschnittsprojekten Q1 und Q3 mit fast allen anderen Teilprojekten. Insbesondere gab es gemeinsame Vorarbeiten mit A1 und B1 für die Fallstudien, über die in Q1 berichtet wird (siehe auch [16]). Im Rahmen dieses Querschittsprojekts wurde ebenfalls in Zusammenarbeit mit B10 die Struktur des prozessunterstützenden Project-Dictionarys entworfen. Mit D2 gab es sehr enge Kontakte auf dem Gebiet der Regelung und Steuerung und bei wärmetechnischen Fragen. Die Zusammenarbeit mit C1 hat sich aus der ersten Fallstudie „Beschreibungstechniken im praktischen Einsatz“ der vorangegangenen Antragsperiode fortgesetzt und zu einem von C1 realisierten Gebäudesimulator geführt, der in mehreren Fallstudien zum Testen eingesetzt wurde. Mit B4 wurden Absprachen über das Kommunikationssystem des Testfelds getroffen, um die in B4 entwickelten Kommunikationsprotokolle einsetzen zu können und B4 Experimente zu ermöglichen.

Eine neue Kooperation mit A3 wird zu einer ersten Implementierung unseres Produktmodells der Methode zum Requirements-Engineering in der Erfahrungsdatenbank führen (siehe auch [18]).

Im Querschnittsprojekt Q3 beteiligte sich D1 an der Entwicklung eines Domänenmodells. D1 übernahm die Rolle des Kunden im Querschnittsprojekt Q4 und arbeitete mit an der Integration des dort verwendeten Prozesses mit der in D1 entwickelten Methode zum Requirements-Engineering.

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] R. Boos. *PEditor.* 1998

[2] T. Disper. *Gebäudeautomation mit CAN auf Basis des Philips P8xC592*. 1999

[3] C. Buch. *TRANSdALL - Entwurf eines Interface-Bausteins zum Transfer des Dallas 1-Wire Bus an eine 8051 kompatible CPU*. 1997

[4] K. Haller. *Development of a Lighting Control System in SDL.* 1999

[5] A. Metzger. *Entwurf einer C Bibliothek zur Ansteuerung von Philips P82C150 SLIO Bausteinen auf Basis des Philips P8xC592 Mikrokontrollers unter dem Echtzeitbetriebssystem KEIL RTX-51.* 1997

Diplomarbeiten

[6] R. Boos. *Erweiterung des Pattern-Konzeptes von PsiGene um Typen und Regeln.* 1999

[7] C. Kaesling. *Domänenspezifischer Software-Architektur-Baukasten mit Design-Patterns.* 1998

[8] B. Noll. *Pattern-Basierte Anwendung Endlicher Automaten in PSiGene*. 1999

[9] A. Metzger. *Tool Supported Prototyping for the Specification and the Design of Building Automation Systems*. 1998

[10] M. Schneider. *Entwurf und Implementierung eines echtzeitfähigen Kommunikationssubsystems für CAN*. 1998

[11] B. Weidmann. *Entwurf und Implementierung eines CAN-Bus-Routers.* 1997

Dissertationen

[12] M. Schütze. *Eine musterbasierte Methode zur domänenspezifischen Modellierung und Generierung von Softwarekomponenten.* Universität Kaiserslautern. Veröffentlicht im Shaker Verlag. 1999

SFB-Berichte

[13] A. Metzger, S. Queins, B. Schürmann. *Installation eines Testraums zur Gebäudeautomation und deren Schnittstelle*. SFB 501-Bericht Nr. 13/96. 1996

[14] S. Queins, G. Zimmermann. *A First Iteration of a Reuse-Driven, Domain-Specific System Requirements Analysis Process*. SFB 501-Bericht Nr. 13/99. 1999

[15] G. Zimmermann. *A Domain Specific Model Architecture for Complex Embedded Systems: A Building Automation Case Study.* SFB 501-Bericht Nr. 10/98. 1998

[16] R. Feldmann, J. Münch, S. Queins, S. Vorwieger, G. Zimmerman. *Baselining a Domain-Specific Software Development Process*. SFB 501-Bericht Nr. 2/99. 1999

[17] S. Queins, B. Schürmann, T. Tetteroo. *Bewertung des dynamischen Verhaltens von SDL-Modellen*. SFB 501-Bericht Nr.9/99. 1999

[18] W. Mahnke, U. Marder, S. Queins, N. Ritter, B. Schürmann. I*ntegrierte Produkt- und Erfahrungsdatenverwaltung*. SFB 501-Bericht Nr. 8/00. 2000

[19] L. Geyer, M. Kronenburg, R. Merz, B. Schürmann. *Domain Modeling in the SFB 501*. SFB 501-Bericht 07/00. 2000

Vorträge / Tagungsbeiträge

[20] J.P. Riegel, C. Kaesling, M. Schütze. *Modeling Software Architecture Using Domain-Specific Patterns.* First Working IFIP Conference on Software Architecture (WICSA 1). Kluwer Academic Publishers. 1999

[21] G. Zimmermann. *Domain Specific Model Architecture for Complex Embedded Systems: A Building Automation Case Study.* H. Franke, B. Kleinjohann, J. Sztipanovits: Adaptation and Evolution in Embedded Information Systems. Seminar No. 98441, Report No. 226. Dagstuhl. 1998

[22] G. Zimmermann. *Modeling embedded digital controllers.* New Information Technologies in Science, Education, Telecommunications and Business (IT+SE'98). Yalta-Gurzuff, Ukraine. 1998

[23] B. Schürmann. *Structure and Design of Building Automation Systems.* New Information Technologies in Science, Education, Telecommunications and Business (IT+SE'98). Yalta-Gurzuff, Ukraine. 1998

[24] M. Schütze, J.P. Riegel, G. Zimmermann. *Pattern-Based Application Generator for Building Simulation.* ESEC 97. Zurich (CH). 1997

[25] F. Heister, J.P. Riegel, M. Schütze, S. Schulz, G. Zimmermann. *Pattern-Based Code Generation for Well-Defined Application Domains.* EuroPLoP '97. Siemens Technical Report 120/SW1/FB. Munich (D) / Zurich (CH). 1997

Zeitschriften / Monografien

[26] M. Schütze, J.P. Riegel, G. Zimmermann. P*SiGene - A Pattern-Based Component Generator for Building Simulation.* Journal Theory and Practice of Object Systems (TAPOS). Vol. 5, No. 2. 1999

[27] J.P. Riegel, M. Schneider. *Dynamic Resource Management for the Controller Area Network.* Erscheint in: CAN Newsletter 6/00, CAN in Automation, Nürnberg, 2000

[28] S. Queins, G. Zimmermann. *A Reuse-Driven, Domain-Specific System Requirements Analysis Process*. Special Issue on Requirements Engineering: The Light Control Case Study of the Journal of Universal Software Engineering. 1999

Online-Dokumente

[29] J.P. Riegel. *PSiGene Pattern Catalog.* 2000 http://wwwagz.informatik.uni-kl.de/projects/psigene.html.

[30] A. Metzger. *Dokumentation der Testinstallation für Gebäudeautomationssysteme*. 2000 http://wwwagz.informatik.uni-kl.de/publications/publications.2.dir/Testraum/paper.ps.gz

[31] A. Metzger. *An Interlink of Building Control System Prototypes and the Lighting Simulation Lumina*. 1999

http://wwwagz.informatik.uni-kl.de/publications/publications.2.dir/Met99.0/paper.ps.gz

6 Fremdveröffentlichungen

[32] W. Frakes, C. Terry. *Software Reuse: Metrics and Models*. ACM Computing Surveys. Vol. 28, No. 2. 1996. S. 415–435.

[33] I. Jacobson, G. Booch, J. Rumbaugh. *The Unified Software Development Process*. Addison-Wesley. 1999

Teilprojekt D2

Modellbasierte Entwicklung wiederverwendbarer Regelungsalgorithmen

Leiter

Prof. Dr.-Ing. habil. L. Litz
geb. 30.8.1949
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Tel.: 0631/205-4050
Fax: 0631/205-4462
litz@eit.uni-kl.de

Mitarbeiter

Ergänzungsausstattung:

Dipl. Ing. J. Schäfer (01.06.98 - 01.5.00)

Dr. rer. nat., Dipl. Phys. R. Merz (seit 01.06.98)

Grundausstattung:

Dipl. Ing. P. Denu

Dipl. Ing. A. Diehl

Dipl. Ing. H. Voos

Techniker M. Behr (bis 8.6.1998), S. Becker (seit 1.7.1999)

1 Kurzfassung der Aufgabenstellung

Das Teilprojekt D2 befasst sich mit der *modellbasierten Entwicklung von Regelungs- und Steuerungsalgorithmen*. Hierbei wurden insbesondere wiederverwendungsunterstützende Konzepte beim Entwurf von Steuerungs- und Regelungsalgorithmen sowie der zugrundeliegenden mathematischen Modelle des zu beeinflussenden technisch physikalischen Prozesses untersucht. Das Ziel war neben der Einbindung der Entwurfsmethodik in den allgemeinen Softwareentwicklungsprozess sowohl eine Erhöhung der Entwurfseffizienz wie auch der Qualität. Einen wesentlichen Anteil innerhalb des Teilprojektes D2 nehmen die Validierungsphasen ein. Sowohl die mathematischen Modelle als auch die Algorithmen sollen auf Komponentenebene, aber auch innerhalb exemplarischer Konfigurationen getestet werden. Als Referenz sollen mittels kommerzieller Gebäudesimulatoren (z.B. TRNSYS) erstellte Simulationsmodelle sowie konkrete mit der entsprechenden Sensorik ausgestattete Testobjekte dienen. Die Idee hierbei ist, durch ausschließliche Anwendung validierter Komponenten aussagekräftige Gesamtmodelle erzeugen zu können.

Neben den eigenen Forschungszielen übernimmt das Teilprojekt D2 innerhalb des SFBs die Rolle des *Domänenexperten*. Dies bedeutet, es unterstützt mit Expertenwissen aus der Anwendungsdomäne Gebäudeautomation, aber auch aus der Steuerungs- und Regelungstechnik. Zum einen erfolgt dies durch die direkte Integration beratender Mitarbeiter in die Teams experimenteller Entwicklungsprojekte, zum anderen durch den Ausbau domänenspezifischer Einträge der Erfahrungsdatenbank. Neben der Entwicklung geeigneter Selektionsmechanismen zur Wiederauffindung des erforderlichen Wissens ist eine gezielte Aufbereitung notwendig, welche die allgemeine Nutzbarkeit ermöglicht.

2 Entwicklung des Teilprojekts in der Förderperiode

Dies ist die erste Förderperiode des Teilprojektes D2. Zum Zeitpunkt der Erstellung dieses Berichtes beträgt die bisherige Gesamtbearbeitungsdauer knapp 2 Jahre, da die beiden Mitarbeiter der Ergänzungsausstattung erst zum 1.6.1998 eingestellt werden konnten. Herr Schäfer, verantwortlich für die Algorithmenentwicklung schied aus dem Teilprojekt zum 1.5.2000 aus. Zusätzlich unterstützt wurde D2 im Bereich formaler Spezifikationen von Steuerungsalgorithmen durch Frau Weng (seit Oktober 99).

D2 beteiligte sich als Infrastrukturprojekt intensiv an den Querschnittsprojekten. Noch im Juni 1998 übernahm D2 im Rahmen des Baseline-Experiments QP 1 *(Prototyping eines Gebäudeautomationssystems)* die Experten- und Kundenrolle. D2 stand beratend bei QP 2 *(Entwurf eines Gebäudeautomationssystems)* zur Verfügung und übernahm in der Fallstudie *RECS (Requirements Engineering Case Study: Light Control)* wieder die Kunden- und Expertenrolle. Eine detaillierte Aufbereitung und Analyse der Anwendungsdomäne folgte im Querschnittsprojekt QP 3 (*Entwicklung eines Domänenmodells*). Es zeigte sich, dass der projektierte zeitliche Umfang der Beteiligung an SFB übergreifenden Querschnittsprojekten von 8h/Woche speziell im Falle von Infrastrukturprojekten weitaus zu niedrig angesetzt war. Jedoch rechtfertigt die Effizienz der teilprojektübergreifenden Querschnittsprojekte den erhöhten Aufwand vollständig. Gerade im Falle von D2 konnte so die Integration des Expertenwissens in den allgemeinen Softwareentwicklungsprozess unter realistischen Bedingungen untersucht werden und praktikable Lösungen gefunden werden.

3 Durchgeführte Arbeiten und Ergebnisse

Die modellbasierte Entwicklung von Regelungs- und Steuerungsalgorithmen erfolgt in den Ingenieurwissenschaften durch Nutzung hochentwickelter, mathematischer Simulationsumgebungen ([24]-[25], [30]-[31]). Diese verfügen über ein umfangreiches Repertoire numerischer Funktionen zum Regelungsdesign, effiziente Simulatoren und Codegeneratoren sowie integrierte Tools zur Analyse und Visualisierung. Direkt kompatible Hardware-in-the-Loop-Anbindungen zum schnellen Prototyping sind meist enthalten. Einige verfügen auch über Bibliotheksverwaltungen zur Unterstützung der Wiederverwendung implementierter Komponenten oder kompletter Modelle. Es existieren jedoch starke konzeptionelle Unterschiede der Simulationssysteme.

So war zunächst die Auswahl einer geeigneten Simulations- und Entwicklungsumgebung zu treffen, die es erlaubt generische, wiederverwendungsunterstützende Entwurfsmethoden in die Modell- und Algorithmenentwicklung einzubringen. In den Ingenieurwissenschaften etablierte Blockschaltbildsimulatoren mit starrer Aufteilung aller Schnittstellen in Ein- bzw. Ausgänge sowie mangelnder Unterstützung von Vererbungs- und Aggregationshierarchien erwiesen sich als ungeeignet.

Anders ist dies bei Systemen zur objektorientierten mathematischen Modellbildung, wie der Simulationsumgebung Dymola. Es zeigt sich, dass gerade die Methode der objektorientierten mathematischen Modellbildung zur Beschreibung technischer Prozesse, charakterisiert durch Klassendefinitionen, Vererbungs- und Aggregationshierarchien, Komponentenkapselung sowie nicht berechnungskausaler Modellierung eine ideale Basis zur Realisierung wiederverwendungsunterstützender Entwurfsverfahren darstellt ([23], [26]-[29]) .

Wesentlich für die Vorgehensweise ist die Kompatibilität der mathematischen Modelle sowie der Algorithmen. Beide müssen zur simulativen Validierung innerhalb eines Gesamtsimulationsmodelles zusammengefügt werden. Abhängig vom Simulationsergebnis muss eine schnelle Modifikation der Reglerauswahl, der Reglerparametrierung und eventuell auch des Detaillierungsgrades des mathematischen Modells erfolgen können.

So wurde sowohl die Bibliothek mathematischer Modelle zur Simulation thermischen Gebäudeverhaltens (3.1 Arbeitsschwerpunkt A) wie auch die hierzu kompatible Bibliothek von Steuerungs- und Regelungsalgorithmen (3.2 Arbeitsschwerpunkt B) in Dymola spezifiziert und implementiert.

3.1 Arbeitsschwerpunkt A

Der Algorithmenentwurf basiert auf einer detaillierten mathematischen Modellierung des dynamischen Verhaltens der zu beeinflussenden Strecke. Dies unterstützt bei der Analyse der Problemstellung, der Detektion der Wirkungszusammenhänge und Verkopplungen und ermöglicht so die Festlegung der eigentlichen Regelkreise. Darüberhinaus erlaubt die Modellierung die Parametrierung der Algorithmen und die Durchführung simulativer Tests zur Überprüfung der Eignung für die konkrete Problemstellung. Auch langsame thermische Prozesse und ihre Beeinflussung, die experimentell nur in langwierigen Messungen verfolgbar wären, können so simulativ in überschaubaren Zeiten untersucht werden.

Startpunkt des Arbeitsschwerpunktes A war die gezielte Auswahl der relevanten physikalischen Prozesse durch Analyse der Anwendungsdomäne „thermisches Gebäudeverhalten“. Es folgte die Untergliederung in Teilprozesse synchron mit der Definition von Schnittstellen. Zur Beschreibung der Teilprozesse wurden Komponentenmodelle entworfen. Transparenz und eine klare Strukturierung wurde erzeugt durch Aufbau einer Aggregationshierarchie, die sich an der echten Strukturierung der physikalischen Realität orientiert. Unterstützt wird dies durch die graphische Implementierung der Komponentenrelationen in hierarchischen Objektdiagrammen.

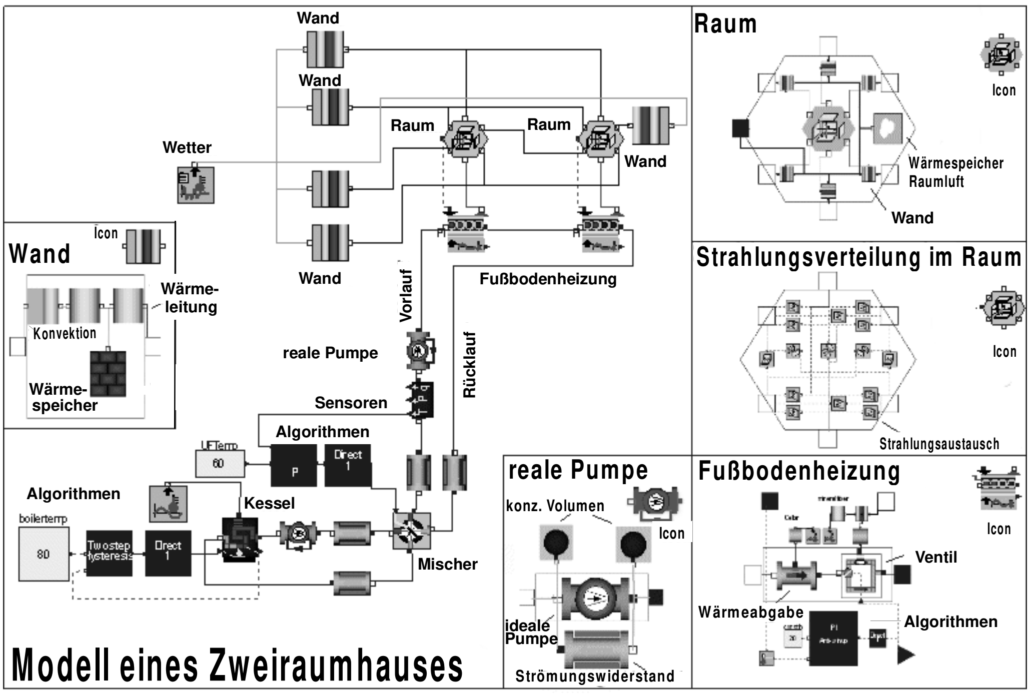

Abbildung 1: Simulationsmodell eines fußbodenbeheizten Zweiraumhauses mit einigen exemplarisch ausgewählten Komponentenmodellen

Innerhalb der Komponentenmodelle wurden physikalische Gleichungen durch Bilanzierung von Erhaltungsgrößen oder auch durch Nutzung phänomenologischer Gesetze erhalten. Beim Entwurf jeder Komponente war abzuwägen, ob diese auf mehrfach nutzbares Basiswissen, beispielsweise physikalische Konstanten und Stoffeigenschaften oder Basisgleichungen zurückgreift, was die Nutzung einer Vererbungshierarchie nahelegt. Die neuen Komponenten verfügen so bei ihrer Definition und Einstellung in die Vererbungshierarchie unmittelbar über die notwendigen physikalischen Grundeigenschaften. Es sind nur noch ihre spezifischen Eigenheiten zu implementieren.

Die Komposition der Komponentenmodelle zum Gesamtsimulationsmodell wurde für exemplarische Konfigurationen typischer Gebäudeteile oder Gebäude vorgenommen. Hierbei erfolgte eine Konzentration auf die verschiedenen physikalisch relevanten Aspekte. Schwerpunkte beim hydraulischen System waren die Wärmeerzeugung im Kessel, das Fließverhalten des Heizungswassers und die Verteilung der Wärme im Rohrleitungssystem, das dynamische Verhalten des Mischventils sowie die Wärmeabgabe von Fußbodenheizung und Radiatoren. Auf Seite der thermischen Speicher wurde die Wärmespeicherfähigkeit sowie der Wärmedurchgang durch bautechnische Komponenten und der Einfluss der Wärmestrahlung betrachtet. Es folgte die Entwicklung von Komponenten zur Wettersimulation und ihre Anbindung an das Hausmodell. Ergänzt zum vollständigen simulationsfähigen Regelkreismodell wurden die Gesamtsimulationsmodelle durch Einbindung von Algorithmenbausteinen aus Arbeitsschwerpunkt B.

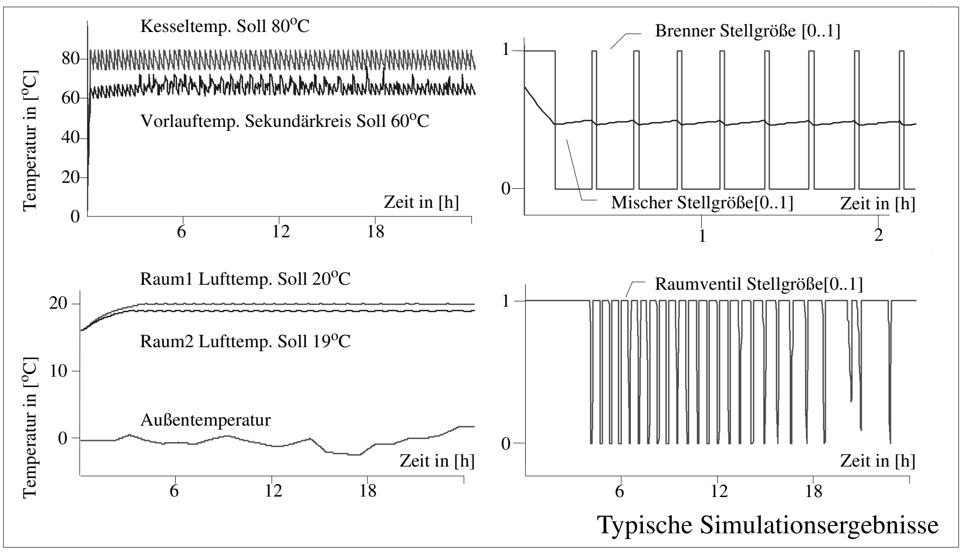

Abbildung 2: Simulationsergebnisse der Regelgrößen sowie Stellgrößen des fußbodenbeheizten Zweiraumhauses aus Abb. 1.

Ausgehend von einfachen, auf einen Aspekt konzentrierten Modellen wurden zunehmend komplexere Gebäudemodelle erstellt (Abb. 1) und simuliert (Abb. 2). Einfache Modelle bilden beispielsweise den Wärmedurchgang durch eine Wand bei festen

Innen- und Außentemperaturen nach. Komplexere Modelle beschreiben Einraum-, Zweiraum- oder Zehnraumhäuser, mit Radiator- und Fußbodenheizung, mehreren Heizkreisen, mischergesteuerten Vorlauftemperaturen, und detaillierten Wettermodellen. Auch die komplexeren Modelle orientieren sich an gebräuchlichen Installationen (Abb. 3).

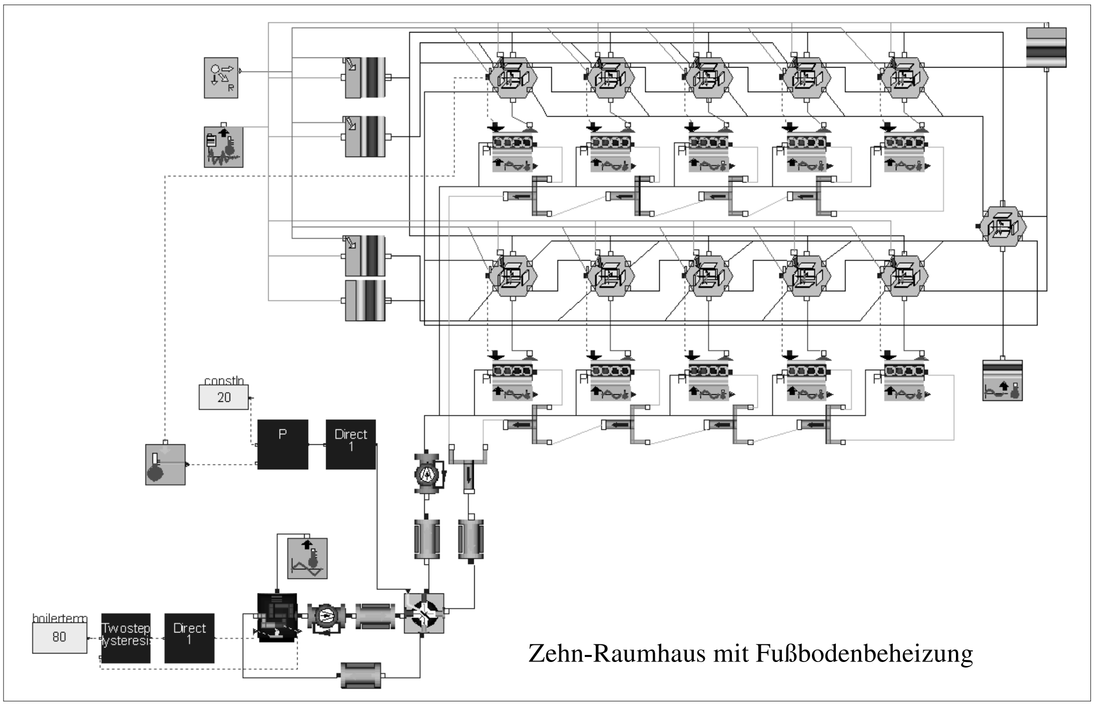

Abbildung 3: Simulationsmodell eines fußbodenbeheizten Zehn-Raumhauses mit hydraulischer Anlage.

Logische Validierungen durch Plausibilitätsbetrachtungen der Ergebnisse, bilanzierende Kontrollrechnungen sowie exemplarische simulative Validierungen wurden angewandt um die Realitätsnähe der mathematischen Modelle zunächst auf Komponentenebene, dann auch für die ausgewählten instantiierten Beispiele zu überprüfen (Abb. 4).

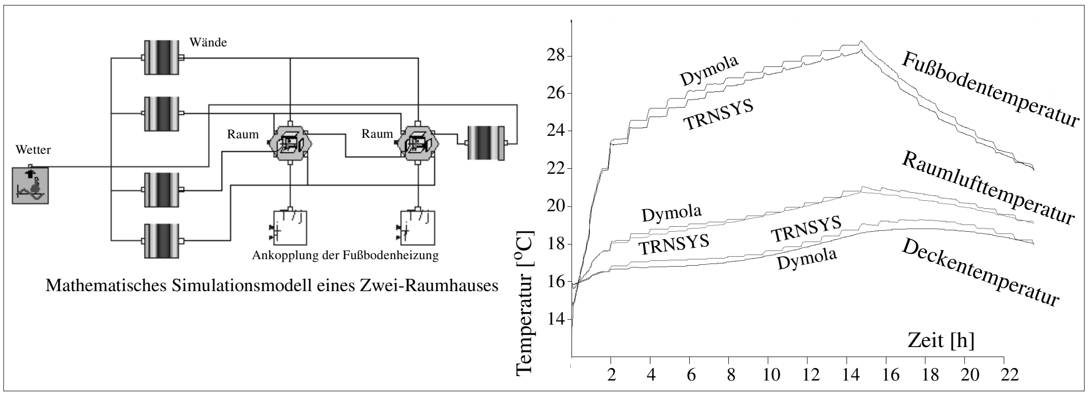

Abbildung 4: Simulative Validierung der mathematischen Modelle in Dymola zur Wärmeverteilung innerhalb eines Raumes durch Vergleich der Resultate mit dem kommerziellen Gebäudesimulator TRNSYS.

Die so validierten Modellkomponenten, wie auch alle ihre Basisklassen wurden ergänzt um Selektionsmerkmale betreffend ihren möglichen Verwendungszweck, Regeln zur Adaption sowie zur Komposition am Einsatzort und in eine Modellbibliothek eingestellt. So kann durch Selektion, Adaption und Komposition für den konkreten Anwendungsfall, generisch ein Gesamtmodell instantiiert werden.

In gleicher Weise werden auch die fertigen, komplexen Modelle in einer Modellbibliothek aufgenommen. Trotz ihrer vollständigen Instantiierung eignen sie sich zur Wiederverwendung. Neben dem exemplarischen Charakter der gespeicherten Modelle kann ein Großteil möglicher neuer Anwendungsfälle durch Komponentenaustausch bzw. -ergänzung der bestehenden Modelle abgedeckt werden. Der modulare Charakter sowie die klare Aggregationshierarchie der entwickelten Modelle unterstützt die einfache Adaption der bestehenden Gesamtmodelle in neuen Einsatzfällen.

Darüberhinaus stellen die Gesamtsimulationsmodelle, ergänzt um die Algorithmenbausteine aus Arbeitsschwerpunkt B lauffähige Regelkreismodelle dar. Sie verfügen damit schon über geeignete Algorithmen bzw. im Falle komplexerer Modelle auch über komplette Regelungskonzepte. Das Auffinden eines geeigneten Simulationsmodells führt damit letzendlich schon zu einem geeigneten Vorschlag bezüglich des Regelungskonzeptes.

3.2 Arbeitsschwerpunkt B

Basierend auf der mathematischen Streckenmodellierung wurden die eigentlichen Regelungsalgorithmen entworfen. Ziel war es mit möglichst wenigen Algorithmen ein möglichst großes Anwendungsspektrum abzudecken. Eine sorgfältige Auswahl der in die Bibliothek einzustellenden Algorithmen war daher unabdingbar. Zunächst wurden Algorithmen auf Komponentenebene, die Teilaspekte und lokale Probleme beherrschen, entwickelt. Mit wachsender Komplexität der mathematischen Modelle aus Arbeitsschwerpunkt A erfolgte dann der Übergang zu komplexeren Regelstrategien, die hierarchisch organisiert, elementare Algorithmen aggregieren.

In gleicher Form wie im Falle der Bibliothek mathematischer Komponentenmodelle existiert eine hierarchisch strukturierte Algorithmenbibliothek. Diese besteht aus Sortimenten von Grundbausteinen zur Sollwertberechnung, Sensor-, Aktorverarbeitung sowie den eigentlichen Reglerkernen (P, PI, PD, PID und Zwei- bzw. Drei-Punkt-Reglern). Zwei exemplarische Tests eines PI-Bausteins sind in Abb. 5 dargestellt.

Diese Bausteine sind ähnlich den Modellkomponenten anwendungsfertig vorkonfektioniert zur Komposition neuer Gesamtalgorithmen. Durch Aufbau mehrfach verschleifter Regelkreise lassen sich durch dieses modulare Konzept auch anspruchsvollere Regelungsaufgaben lösen. Gerade verkoppelte Prozesse mit stark unterschiedlicher Eigendynamik, wie sie im Bereich thermischen Gebäudeverhaltens häufig auftreten, erfordern dies.

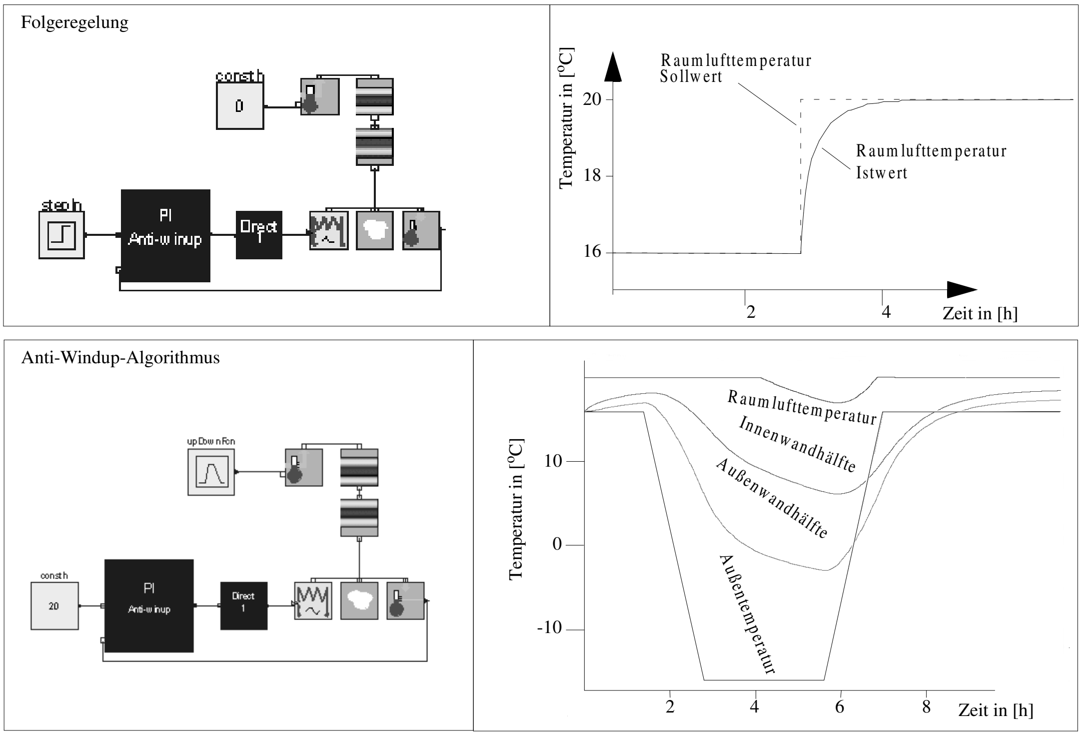

Abbildung 5: Test eines PI-Reglerbausteines mit Anti-Windup-Algorithmus. Der Regler geht auf die maximal mögliche Stellgröße, verbleibt auf diesem Wert, reagiert jedoch korrekt beim erneuten Ansteigen der Temperatur.

Die generische Entwurfsmethodik basiert auf der Modularität des Konzeptes begleitet von dem sukzessiven, prozessbegleitenden Ausbau der Algorithmenbibliothek. Das zur Verfügung stehende Spektrum gespeicherter, validierter Algorithmenkomponenten sowie beispielhafter komplexer Algorithmen soll auch weiterhin stetig zunehmen. Die Algorithmenbibliothek beinhaltet neben den eigentlichen Algorithmenbausteinen Regeln zu ihrer Auswahl, ihrem Einsatzgebiet und ihrer Verknüpfung.

Darüberhinaus existieren konkrete Anwendungsbeispiele durch die in Arbeitsschwerpunkt A schon aufgeführten Bibliothekseinträge der Modellbibliothek mit bereits vollständigen, entwickelten Modellen. Die hier abgelegten Modelle entsprechen simulationsfähigen Regelkreismodellen. Sie bestehen aus instantiierten Gesamtsimulationsmodellen des zu beeinflussenden technisch physikalischen Prozesses, der sogenannten Strecke, sowie den zugehörigen Algorithmen. Sie sind simulativ und exemplarisch experimentell validiert. Sie repräsentieren typische Anwendungsbeispiele von Regelungs- und Steuerungsalgorithmen, die zur Orientierung bei einer Neuentwicklung dienen können, die aber auch durch Modifikation und Adaption an eine neue Aufgabenstellung anzupassen sind. Beispiele unterschiedlichen Detaillierungsgrades und Komplexität wurden entwickelt. Die Bibliothek wurde projektbegleitend durch Aufnahme neuer Modelle erweitert. Die graphische Implementierung sowie der modulare Aufbau der mathematischen Modelle ermöglichen es dem Domänenexperten sofort ein intuitives Bild sowohl der beschriebenen Strecke, als auch der angewandten Lösungskonzepte zu erhalten. Zudem erläutern Kommentare die Charakteristika der durch die Modelle mathematisch beschriebenen, realen technisch physikalischen Prozesse bzw. ihrer Charakteristika.

Alle bislang beschriebenen Bibliotheken enthalten in einer konkreten Simulationssprache implementierte, durch Kommentare ergänzte mathematische Beschreibungen. Sie unterstützen vorwiegend den Domänenexperten beim Entwurf der Steuerungs- und Regelungsalgorithmen (siehe. Abb. 6), da zumindest eine gewisse Erfahrung mit dem mathematischen Simulationstool, sowie dem Begriff des Regelkreises zum Verständnis unabdingbar ist. Neben diesen in mathematischer Form spezifizierten Modell- und Algorithmenbibliotheken entwickelt D2 als Essenz formal spezifizierte, projektübergreifend nutzbare Regelungs- und Steuerungsalgorithmen. Beim Design wurde großen Wert darauf gelegt, dass die Algorithmen auch für Softwareentwickler nutzbar sind und direkt in allgemeine Softwareentwicklungsprozesse einfließen können. Diese zweite Kategorie teilprojekteigener Ergebnisse ist die eigentliche Schnittstelle zum SFB. Sie umfasst einsatzbereite, implementierte Algorithmen die vom Softwareentwickler in das Gesamtsystem integriert werden können. Diese werden letzendlich aus den formal spezifizierten mathematischen Beschreibungen der Simulationssprachen generiert. Bis zum Abschluss der Förderungsperiode soll die automatische Umsetzung vollständig möglich sein. Der Experte liefert dann letztendlich aus der ihm übergebenen Anforderungsbeschreibung der regelungstechnischen Aufgabe sowie der Beschreibung des technisch physikalischen Prozesses integrierbare black-box-artige Codefragmente. Insbesondere bei komplexen regelungstechnischen Aufgaben zur Steuerung großer Systeme, die über die Anwendung einiger weniger Grundbausteine hinausgehen, ist eine stärkere Integration in den allgemeinen Entwicklungsprozess mit der Folge einer aus Sicht der Softwareentwickler erhöhten Transparenz wünschenswert. Während auf niedrigen automatisierungstechnischen Ebenen die Bausteinanwendung adäquat ist und den Softwareentwickler von numerischen, mathematischen Überlegungen befreit, sind auf höheren Ebenen andere Konzepte erforderlich. Daher erarbeitete das Teilprojekt D2 eine ergänzende Methodik.

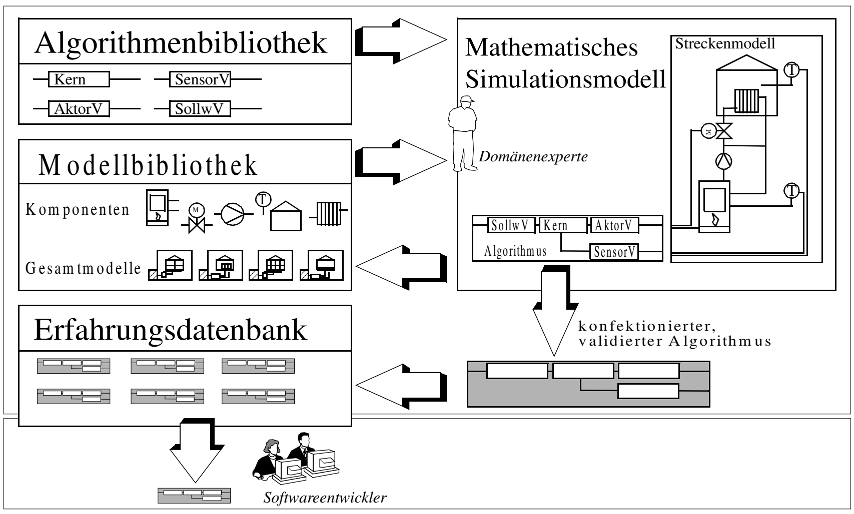

Abbildung 6: Teilprojekteigene Bibliotheksstruktur und Integration im Gesamt-SFB

Übergeordnete Strategien lassen sich auf einer höheren Abstraktionsebene leicht spezifizieren und können im Prinzip als Erweiterung der initialen Anforderungsbeschreibung aufgefasst werden. Ein Beispiel hierfür ist die außentemperaturabhängige Nachführung der Vorlauftemperatur entsprechend einer Heizkennlinie, ergänzt durch thermostatische Einzelraumregelung. Verborgen im mathematischen Simulationsmodell oder im implementierten Codefragment bleibt diese Strategie dem Softwareentwickler bei Fehlersuche oder Notwendigkeit der Modifikation ebenfalls verborgen. Eingeflossen in einer frühen Phase des allgemeinen Softwareentwicklungsprozesses als zusätzliche Anforderung aus Expertensicht erfolgt die Implementierung transparent als Teil des Gesamtsystems. Das Teilprojekt erarbeitet daher zusätzlich einen Katalog höherer Strategien, Taktiken und Konzepte, die formal spezifiziert in einer eigenen Untereinheit der Erfahrungsdatenbank zur Verfügung gestellt werden sollen.

Von Bedeutung sind hierbei insbesondere Selektionsmaße zum Auffinden der Einträge in der Erfahrungsdatenbank. Das Auffinden geeigneter Lösungen basiert auf einer formalen Definition der Eigenschaft *'geeignet'*. Eine Lösung bzw. auch Komponente ist geeignet, falls sie ursprünglich aufgrund *'ähnlicher'* Anforderungen entwickelt wurde. Als Ähnlichkeitsmaßdient dient zunächst die hierarchische Ebenenstruktur der Automatisierungstechnik. Auf niedrigen Ebenen findet man mathematische Beschreibung, auf höheren Ebenen Ergänzungen der initialen Anforderungen aus Expertensicht. Kriterien der Detailauswahl folgen dann aufgrund der Kenntnis der Art und Anzahl von Aktoren und Sensoren, den Charakteristika der konkreten Strecke sowie der Anforderungsdokumente.

4 Zusammenarbeit mit anderen Teilprojekten

Als Infrastruktur- und Serviceprojekt ergab sich eine Vielzahl von Kooperationen mit anderen Teilprojekten.

D2 übernahm im Rahmen des Querschnittprojektes *(QP 1: Prototyping eines Gebäudeautomationssystems)* in einer Doppelbesetzung, zunächst gemeinsam mit D1 die Kunden- und zugleich Expertenrolle. Es war hierbei für die Erstellung von Problembeschreibungen, dem Entwurf und Design geeigneter Steuerungs- und Regelungsalgorithmen zur Integration in das Gesamtsystem, sowie für die Entwicklung realistischer Testszenarien verantwortlich. Gerade die Interaktion mit den Softwareentwicklern innerhalb des Teams erwies sich als idealer Startpunkt des Teilprojektes zur Definition der Expertenrolle und ihrer Aufgaben innerhalb des SFBs. Wann welches Wissen wo in den Entwicklungsprozess einfließen muss, trat am konkreten Beispiel direkt in Erscheinung. Es zeigte sich hierbei auch die primäre Notwendigkeit eine gemeinsame Sprache zu finden, aber auch eine klare Abgrenzung der Tätigkeitsgebiete zwischen Domänenexperte und Softwareentwicklern festzulegen.

Begleitend zu der Teamarbeit wurde in einer Analysephase das notwendige Expertenwissen bezüglich der bauphysikalischen Zusammenhänge, der technischen Installationen sowie Vorschriften und DIN-Normen zusammengestellt. Die Selektion des notwendigen Wissens sowie die allgemeinverständliche Aufbereitung schloss sich den Recherchen an. Dieses Wissen floß dann in das zusammen mit B10 („Anwendungsentwicklung mit vorkonfektionierten Softwaresystemen“), C1 („Formale Beschreibungstechniken“) und D1 („Anwendungssystem Gebäude“) durchgeführte Querschnittsprojekt QP3 (*QP 3: Entwicklung eines Domänenmodells*) ein. Dieses Domänenmodell soll als Referenzmodell bei der Erstellung von Systemfamilien dienen, es stellt aber auch einen Schritt zur Entwicklung eines generischen Anforderungsdokumentes dar. Eine weitere Detaillierung wird sich anschließen. So kann das Domänenmodell letztendlich dazu dienen, detailliertes Expertenwissen zu speichern und bei der Instanziierung eines konkreten Anforderungsdokumentes den Domänenexperten mit allen im konkreten Anwendungsfall regelungstechnisch relevanten Sachverhalten zu versorgen.

D2 beteiligte sich als Kunde und beratender Domänenexperte an der von den Teilprojekten B4 („Generische Kommunikationssysteme“) und C1 („Formale Beschreibungstechniken“) durchgeführten Fallstudie *„Requirements Engineering Case Study: Light Control"*. D2 erstellte die initiale, informelle Problembeschreibung und beantwortete in der Kundenrolle alle Fragen, die bei der Erstellung der formalen Anforderungsspezifikation auftraten. Gerade hier war die Integration von D2 als Domänenexperte notwendig, um regulierend ein unbeabsichtigtes Divergieren des erforderlichen Lösungsaufwandes zu vermeiden und der Studie eine realitätsnahe Basis zu geben. Ergebnisse der Studie werden in einem Sonderheft des Journals of Universal Computer Science, Springer Heidelberg veröffentlicht werden. Zudem beteiligte sich D2 als regulärer Teilnehmer an der Fallstudie mittels eines eigenen methodenorientierten Beitrags zur formalen Spezifikation des Steuerungsalgorithmus durch interpretierte Petri-Netze *(submitted to J.UCS, special issue: „Requirements Engineering Case Study Light Control“).*

Ein wesentlicher Schwerpunkt des Teilprojektes D2 ist die Nutzbarmachung der teilprojekteigenen Ergebnisse bzw. Produkte. Die Einlagerung der teilprojekteigenen Ergebnisse in die Gesamterfahrungsdatenbank des SFB wurde zusammen mit A1 („SE-Labor“) und A3 („Objektrelationale Datenbanktechnologie“) erarbeitet. Die Einlagerungen spiegeln die zwei Kategorien der Ergebnisse wieder. Während die zur Simulation geeigneten mathematischen Beschreibungen vor allem von den Domänenexperten genutzt und wiederverwendet werden, werden die vorkonfektionierten Algorithmen von den Softwareentwicklern als fertige Module in Softwarelösungen eingesetzt. Der Schwerpunkt lag daher insbesondere bei der Ausarbeitung charakterisierender Attribute zur Selektion gerade der vorkonfektionierten Algorithmen. Hinzu kommen die formal spezifizierten Regeln und Strategien.

5 Eigene Veröffentlichungen im Berichtszeitraum

Projektarbeiten

[1] C. Anwander: *Simulation des thermischen Verhaltens eines Niedrigenergiehauses mit dem Gebäudesimulator TRNSYS*; Projektarbeit, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, Januar 2000

[2] M. Grünewald: *Entwicklung eines Industrie PID-Reglers unter vergleichender Anwendung der Tools SDT*, Matlab/Simulink und Dymola, Juli 2000

[3] F. Felgner: *Objektorientierte mathematische Modellierung der Hydraulik von Heizungsanlagen,* September 2000.

Diplomarbeiten

[4] O. Seel: *Objektorientierte mathematische Modellbildung*; Diplomarbeit, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, Juni 2000

[5] E. Sitompul: *Entwicklung von generischen Regelungsalgorithmen zur Hausbeheizung unter kombinierter Anwendung von Dymola und Matlab,* September 2000

Dissertationen: keine im Berichtszeitraum

SFB-Berichte

[6] R. Merz und L. Litz: *Anwendung generischer Methoden bei der objektorientierten Modellierung großer, komplexer dynamischer Systeme*, Technical Report SFB 501- 07/99, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, 1999

[7] J. Schäfer und L. Litz: *Objektorientierte Entwicklung von Regelungs- und Steuerungsalgorithmen für die Anwendungsdomäne Gebäudeautomation unter den Aspekten Wiederverwendbarkeit und Generik*, Technical Report SFB 501-010/99, Fachbereich Elektro- und Informationstechnik, Universität Kaiserslautern, 1999

Vorträge: keine im Berichtszeitraum

Tagungsbeiträge

[8] A. Diehl, M. Becker: *Ein dezentrales, modulares Konzept zur Automatisierung kältetechnischer Anlagen.* In: Tagungsband GMA-Kongreß'98 Mess- und Automatisierungstechnik *"Neue Entwicklungen Technologie Anwendungen"*, Ludwigsburg, Juni 1998. VDI Berichte 1397, S. .395-402, VDI-Verlag, Düsseldorf, 1998

[9] A. Diehl, L. Litz: *Object-Oriented Top-Down Modeling For Dynamic Simulation Of Copression Plants*. In the Proceedings of the 20th International Congress of Refrigeration, IIR/IIF, Sydney, 1999, Paper Code: 064

[10] A. Diehl: *Dynamic Simulation of Refrigeration Plants Based on a Object-Oriented Top-Down Model.* In: Proceedings of Modelling and Control Conference 2000, MIC-2000, Innsbruck, 2000, Paper Code: 306-021

[11] G. Frey und L. Litz: *Entwurf und formale Verifikation von Steuerungen mit interpretierten Petri-Netzen*, In: Tagungsband GMA-Kongreß'98 Mess- und Automatisierungstechnik *Neue Entwicklungen Technologie Anwendungen*, Ludwigsburg, Juni 1998. VDI Berichte 1397, S. 291-298, VDI-Verlag, Düsseldorf, 1998.

[12] G. Frey: *Simulation of Hybrid Systems based on Interpreted Petri Nets*, Proceedings of the IEE International Conference on Simulation - Innovation Through Simulation, York, 30.09. - 02.10.98, S. 168-175.

[13] G. Frey und L. Litz: Verification and Validation of Control Algorithms by Coupling of Interpreted Petri Nets, Proceedings of the IEEE SMC'98, San Diego, 11.-14. 10.98, Band 1, S. 7-12

[14] J. Schäfer: *Reuse- and Object-Oriented Development of Standard Algorithms* IFAC Workshop on Digital Control: *Past, Present and Future of PID Control* 5.-7.4.2000 Vapor Universitari, Terassa, Spain 2000

[15] H. Voos, R. Habtom, L. Litz: *Development of an Engineering Tool for the Design and Optimization of Knowledge-based Control Structures*. In Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering SEKE'99, Kaiserslautern 1999, S. 327-331

[16] X. Weng, L. Litz: *Verification of logic control design using SIPN and model checking - Method and case study.* Erscheint in: Proceedings of the American Control Conference ACC 2000, Chicago, 28.-30. Juni 2000.

Zeitschriften / Monografien

[17] M. Becker und A. Diehl: *Optimierung von Kälteanlagen*. In: *ki Luft- und Kältetechnik* 5(1998), C. F. Müller Verlag, Hüthig GmbH, Heidelberg, S.262-263

[18] M. Becker, A. Diehl, R. Habtom: *Modellierung einer kältetechnischen Anlage mit rekurrenten Neuronalen Netzen.* atp - Automatisierungstechnische Praxis 9/99, R. Oldenbourg Verlag, S. 22-31.

[19] A. Diehl, M. Becker, L. Litz: *Die Automatisierung in der Kältetechnik - Gestern und heute.* In: ki Luft- und Kältetechnik 2(1999); C.F.Müller Verlag, Hüthig GmbH, Heidelberg; S.101 - 106.

[20] A. Diehl: *Technische und wirtschaftliche Optimierung von Kälteanlagen. In: ki Luft- und Kältetechnik* 5(1999); C.F.Müller Verlag, Hüthig GmbH, Heidelberg; S.296 - 297

[21] R. Merz und L. Litz: *Objektorientierte mathematische Modellierung: Generische Methoden bei komplexen dynamischen Systemen*, Informatik Spektrum 2(2000), Springer Verlag Heidelberg, S. 90-99

[22] L. Litz und G. Frey: *Methoden und Werkzeuge zum industriellen Steuerungsentwurf - Historie, Stand, Ausblick,* at-Automatisierungstechnik 47(1999)4, R. Oldenbourg Verlag, S. 145-156

6 Fremdveröffentlichungen

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

[24] *Dymola.* Lund/Schweden: Fa. Dynasim, 1999; *http://www.dynasim.se*

[25] J. Hoffmann: *Matlab* und *Simulink*. Bonn: Addison -Wesley 1998

[26] M. Otter: Objektorientierte Modellierung physikalischer Systeme Teil 1-5. at1, A1-A4(1999); at2, A5-A8(1999); at3, A9-A12(1999); at4, A13-A16(1999); at5, A17-A20 (1999)

[27] M. Otter, B. Bachmann: Objektorientierte Modellierung physikalischer Systeme Teil 6. at6, A21-A24(1999)

[28] M. Otter, H. Elmqvist, L. Mattsson und S.E. Mattsson: *Objektorientierte Modellierung physikalischer Systeme* Teil 7 + 8. at9, A25-A28(1999); at11, A29-A32(1999)

[29] M. Otter, C. Schlege: *Objektorientierte Modellierung physikalischer Systeme* Teil 9. at12, A33-A36(1999)

[30] *Simulink*: *Matlab/Simulink*, Natick/USA MathWorks Inc.1999; *http://www.Mathworks.com*

[31] *Smile*; *http://buran.fb10.tu-berlin.de/Energietechnik/EVT_KT/smile/* *Smile*-Usertag GMD first, Berlin: 16. April, 1999; *http://www.first.gmd.smile*

Querschnittsprojekt Q1

Durchführung eines domänen-spezifischen Requirements-Engineering Prozesses

Leiter

Prof. Dr. Gerhard Zimmermann (D1)

Mitarbeiter

Dipl. Inform. Raimund L. Feldmann (A1)

Dipl. Inform. Stefan Vorwieger (A1)

Dipl. Inform. Wolfgang Mahnke (A3)

Dipl. Inform. Jürgen Münch (B1)

Dipl. Inform. Christian Peper (B4)

Dipl. Inform. Martin Becker (B5)

Dipl. Inform. Martin Kronenburg (C1)

Dipl. Inform. Stefan Queins (D1)

Dipl. Ing. Jürgen Schäfer (D2)

Dr. Rolf Merz (D2)

1 Einleitung

In diesem Querschnittsprojekt wurden zwei Fallstudien mit einem Team von Mitarbeitern aus mehreren Teilprojekten durchgeführt, um Messungen zur Effizienz und Produktqualität an größeren Beispielen in einer Gruppe von Entwicklern und Experten durchführen zu können. Die Fallstudien (Case-Studies), die im weiteren CS3/1 und CS3/2 genannt werden, wurden im SE-Labor des SFB 501 mit Unterstützung der Teilprojekte A1 und B1 durchgeführt. Aufgabenstellung, Leitung und Prototyping wurde von dem Teilprojekt D1 übernommen, wobei ein Gebäudesimulator von Teilprojekt C1 als Ergebnis einer früheren Fallstudie zur Verfügung gestellt wurde.

Beide Fallstudien hatten das Ziel, ausgehend von einer Problembeschreibung der Domäne Gebäudeautomation, ausführbare System-Requirements zu erstellen und sie zu validieren und zu verifizieren. Dazu sollte nach einer in D1 entwickelten domänenspezifischen Analysemethode (siehe auch Ergebnisbericht D1, Arbeitsschwerpunkt E) vorgegangen werden und mit von Teilprojekt A1 definierten Messmethoden begleitet werden. Als Notationen wurden in beiden Fällen informelle Beschreibungen für die Problembeschreibung und die ersten Typbeschreibungen, UML für die Objektstruktur und SDL für die ausführbaren Modelle verwandt. CS3/1 wurde als Baseline angesehen, da in dieser Domäne keine Messergebnisse vergleichbarer Analysemethoden gefunden werden konnten. CS3/2 ist eine erste Iteration der verwandten Methode, wobei in CS3/1 erkannte Mängel abgestellt und zusätzliche Verbesserungen erprobt wurden. Dies bezieht sich vor allem auf die konsequente Nutzung von Wiederverwendung einzelner Artifakte und ein geändertes Vorgehensmodell. In CS3/2 musste außerdem die Aufgabenstellung modifiziert werden, da durch die Kenntnisse der Entwickler über den CS3/1-Entwurf die Messungen sonst erheblich verfälscht worden wären. Im Prinzip ist durch diese Maßnahme ein Vergleich der Messdaten zwischen den beiden Fallstudien möglich. Allgemeingültige Aussagen sind beim Vergleich von zwei Fallstudien natürlich nicht möglich, aber ein Trend der Daten konnte festgestellt werden, der sich z.B. bei der Wiederverwendung mit der Erfahrung anderer deckt. Dadurch konnte die Messmethode als richtig angesehen werden.

2 Fallstudie CS3/1

Die Aufgabenstellung von CS3/1 war es, eine Lichtsteuerung für ein Stockwerk in Gebäude 32 der Universität Kaiserslautern, in dem auch das Testfeld des Teilprojekts D1 liegt, zu entwerfen. Diese Aufgabenstellung wurde durch insgesamt 39 Needs notiert, die in drei Gruppen (User-, Facility-Manager-, Nonfunctional-Needs) eingeteilt wurden. Es wurde nicht vorausgesetzt, dass diese Liste vollständig und konsistent war, da dies auch von einem realen Kunden nicht verlangt werden kann. Im Querschnittsprojekt Q4 wurde eine Fallstudie durchgeführt, die diese Eigenschaften garantieren sollte. zusätzlich zu den Needs existierte eine Beschreibung des Gebäudes mit den bereits installierten Sensoren und Aktuatoren. Beides zusammen waren die Ausgangsdokumente der Fallstudie. Die Vorgehen war durch ein Modell beschrieben, das von den Entwicklern strikt eingehalten werden musste. Desweiteren war ein Mess-und Testplan festgelegt, um Aufwands-, Kalenderzeit- und Fehlerdaten zu erfassen. Als Rechnerunterstützung wurde eine auf RCS basierende Umgebung bereitgestellt.

Insgesamt wurden 25 verschiedene Objekttypen modelliert, woraus sich 392 Instanzen ableiteten. Der Aufwand des gesamten Durchlaufs betrug 19000 Minuten (ca. 8 Personenwochen). Um einen Vergleich mit nachfolgenden Projekten zu ermöglichen, wurde eine Größe eingeführt, die die Komplexität eines Projekts beschreibt. Sie errechnet sich im wesentlichen aus den Aufgaben, die die einzelnen Objekttypen zu erfüllen haben. Diese Komplexität errechnete sich für CS3/1 zu 688. Der Quotient aus dem Aufwand und dieser Größe bestimmt die Produktivität. Hier beträgt sie 27,6.

Einzelheiten der Analysemethode, der Experimentierumgebung und der Auswertung sind in [1] dargestellt.

Als wesentliche Erfahrungen sollen nur die folgenden genannt werden:

- Ein Team der Größe von sieben Entwicklern kann eine in dieser Domäne typische Aufgabe in vernünftiger Zeit lösen.
- Inspektion und Prototyping haben sich gut ergänzt, um unterschiedliche Fehlertypen in unterschiedlichen Dokumenttypen zu finden.
- Durch eine von sehr großen Projekten übernommene sehr strikte Phaseneinteilung des Prozesses entstanden unnötige Wartezeiten und kleine Verbesserungen führten zu aufwendigen Iterationen.
- Fehler traten vor allem an den Schnittstellen zwischen Objekttypen auf, was auf eine inkonsequente Führung des Project-Dictionaries zurückzuführen war.
- Wiederverwendung war auf die Templates beschränkt und könnte erweitert werden.
- Prototyping und damit eine Validierung war erst am Ende der Analyse möglich.

Aus diesen Erfahrungen wurde als Konsequenz eine Reihe von Änderungen an dem Prozess eingeführt, die eine effizientere Durchführung gewährleisten sollten. Dies wurde in einer zweite Fallstudie (CS3/2) validiert.

3 Fallstudie CS3/2

In CS3/2 wurde die Problembeschreibung von CS3/1 um eine Temperatursteuerung ergänzt. Es kamen insgesamt 30 neue Needs hinzu. Um bessere Aussagen über die Wiederverwendbarkeit von Artefakten machen zu können, wurde festgelegt, dass der CS3/1 Entwurf nicht als Ganzes, sondern nur in Form kleinerer Einheiten wiederverwandt werden sollte. Dazu wurden einzelne Artefakte generalisiert und in einer Reuse-Library abgelegt.

Anstelle der strikten Phaseneinteilung wurde als Vorgehensmodell ein wesentlich flexiblerer Verfeinerungsansatz gewählt. Dadurch konnten alle Dokumente ständig verfeinert und korrigiert werden, ohne aufwendige Iterationen zu durchlaufen.

Weiterhin wurde das Project-Dictionary durch eine Datenbank (siehe [3]) realisiert und die ständige Wartung erzwungen. Dadurch verringerten sich die Schnittstellenfehler drastisch.

Bei jedem neu zu erstellenden Artefakt wurde die Suche in der Reuse-Library erzwungen. Gründe für die Nutzung bzw. Ablehnung der gefundenen Reuse-Artefakte und der Aufwand für die Anpassung wurden detailliert erfasst. Die berechnete Produktivitätssteigerung von 1,22 ist zwar typisch, ist aber jedoch aufgrund der relativ kleinen Anzahl von potenziellen Wiederverwendungen noch ungenau.

Durch die Aufteilung der Objektstruktur in kleinere Teilgraphen und die dynamische Zuweisung derselben zu verschiedenen Entwicklern entfielen die Wartezeiten. Durch die Nutzung von Stubs und Treibern konnten die verschiedenen Teile des Systems frühzeitig und getrennt voneinander getestet werden.

Die Anzahl der Objekttypen erhöhte sich um 12 auf 37 Objekttypen, die zu 947 Objekten instanziiert wurden. Die Gesamtkomplexität betrug 1826. Der Gesamtaufwand betrug 29000 Minuten (ca. 12 Personenwochen), woraus sich eine Produktivität von 15,9 ergibt.

Der vollständige Ergebnisbericht über CS3/2 findet sich in [2].

4 Zusammenfassung

Die Änderungen wurden von den Entwicklern subjektiv als Verbesserung bewertet und sie ergaben insgesamt eine Produktivitätssteigerung von 1,73. Die Anzahl der Fehler und nötigen Änderungen (relativ zu den Gesamtkomplexitäten) ging auf ca. die Hälfte zurück. Jedoch ist der durchschnittliche Aufwand zur Fehlerbeseitgung gestiegen, da in erster Linie die Fehlerquellen beseitigt wurden, die einen relativ einfachen Fehler verursachen.

Beide Fallstudien haben sich als sehr wertvoll für die quantitative Erprobung der Analysemethode und der Messmethode herausgestellt und den Aufwand, der von den Mitarbeitern in die Teams eingebracht wurde, voll gerechtfertigt. Die für diese Fallstudien entwickelten Problembeschreibungen waren darüberhinaus auch Grundlage weiterer Fallstudien (siehe Querschnittsprojekt Q2). Weiterhin steht mit den entwickelten System-Requirements eine Eingabe für die nachfolgenden Entwicklungsphasen, speziell der Design-Phase, zur Verfügung.

5 Eigene Veröffentlichungen im Berichtszeitraum

SFB-Berichte

[1] R. Feldmann, J. Münch, S. Queins, S. Vorwieger, G. Zimmermann: *Baselining a Domain-Specific Software Development Process*, SFB 501 Bericht 02/99, Fachbereich Informatik, Universität Kaiserslautern, 1999

[2] S. Queins, G. Zimmermann: *A First Iteration of an Reuse-Driven Domain-Specific System Requirements Analysis Process*, SFB 501 Bericht 13/99, Fachbereich Informatik, Universität Kaiserslautern, 1999

[3] L. Baum, M. Becker, L. Geyer, G. Molter: *Zwei Unterstützungswerkzeuge auf dem Gebiet domänspezifischer Softwareentwicklung*, SFB 501 Bericht 14/99, Fachbereich Informatik, Universität Kaiserslautern, 1999

Zeitschriften / Monographien

[4] S. Queins, G. Zimmermann: *Reuse-Driven, Domain-Specific System Requirements Analysis Process*, erscheint in *Special Issue on RequirementsEngineering: The Light Control Case Study*, Journal of Universal Computer Science, 2000.

Ergebnisbericht Querschnittsprojekt Q2

Entwicklung eines Gebäudeautomationssystems

Leiter

Prof. Dr. Dieter Rombach

Mitarbeiter

Ergänzungsausstattung:

Dipl.-Inform. Raimund L. Feldmann
Dipl.-Inform. Jürgen Münch
Dipl.-Inform. Stefan Vorwieger

Grundausstattung:

Dipl.-Inform. Natalie Ardet
Dipl.-Inform. Antje von Knethen

Zusammenfassung

Im Rahmen des SFB 501 werden alle verfügbaren Techniken, Methoden und Werkzeuge regelmäßig auf ihre positive Wirkung in Bezug auf Qualität, Kosten und Zeit bei der Entwicklung umfassender Gebäudeautomationssysteme getestet. Dabei wird zum einen die Wirkung neu entwickelter Techniken, Methoden und Werkzeuge unter möglichst realistischen Bedingungen getestet, zum anderen werden Baseline-Erfahrungen erfasst, gegenüber denen Ergebnisse zukünftiger Forschungsarbeiten als Verbesserungen demonstriert werden müssen. Im Querschnittsprojekt Q2 wurde die erste umfassende Baseline (V1) erstellt. Dabei wurden bis Mitte 1999 verfügbare und isoliert erprobte Techniken und Werkzeuge eingesetzt. Im Rahmen eines Teamprojekts mit Informatikstudenten im Hauptstudium wurde ein komplettes Gebäudeautomationssystem entwickelt. Die dabei gewonnenen Erfahrungen, wie das verwendete Prozessmodell sowie Aufwands-, Zeit- und Qualitätsmodelle, wurden als Referenzen für zukünftige Baselines konserviert.

1 Bedeutung von Entwicklungsbaselines im SFB 501

Der positive Beitrag einer Technik oder eines Werkzeugs (z. B.: Reduktion des Entwicklungsaufwands, Verbesserung der Qualität) unterliegt in praktischen Entwicklungensprojekten sehr vielen Einflüssen (z. B.: Erfahrung der Entwickler, Einhaltung des Prozesses, Robustheit der Entwicklungsplattform). Darüber hinaus können sich unterschiedliche Techniken/Werkzeuge gegenseitig beeinflussen. Deshalb ist es nicht ausreichend, neu entwickelte Techniken und Werkzeuge isoliert empirisch zu untersuchen, sondern sie müssen in geeigneter Kombination (definiert über ein Prozessmodell) erprobt werden. Solche notwendigen Scale-Up-Experimente, sogenannte Baselining-Entwicklungen, definieren Baselines in Bezug auf den praktischen Effekt einer neuen Kombination einzelner Techniken/Werkzeuge. Baselining-Entwicklungen sind teuer und können nur in größeren Abständen wiederholt werden, da komplexe Systeme aus der Anwendungsdomäne des SFB vollständig entwickelt werden und somit eine Vielzahl von personellen Ressourcen (für die Planung, Entwicklung, Messdatenerfassung, Analyse etc.) über einen längeren Zeitraum gebunden ist. Aus diesem Grund wird an die eingesetzten Methoden, Techniken und Werkzeuge die Bedingung gestellt, dass sie bereits im Verantwortungsbereich der jeweiligen Teilprojekte im Rahmen von Technologie-Erprobungen bzw. Technologie-Experimenten erfolgreich erprobt wurden. Die Resultate von Baselining-Entwicklungen sowie die hierbei gemachten Erfahrungen sollen wiederum Änderungen oder Neuentwicklungen von Methoden, Techniken und Werkzeugen durch die Teilprojekte motivieren, die zu Verbesserungen führen.

2 Ziele von Q2

Ziel des Querschnittsprojekts Q2 war die Entwicklung einer kompletten Baseline von Modellen und Erfahrungen für den bis Mitte 1999 verfügbaren Satz von Beschreibungstechniken, Entwicklungsmethoden und Werkzeugen. Diese Techniken, Methoden und Werkzeuge beinhalten die für OO-Entwicklung gängigen

- Beschreibungstechniken für die Erstellung der Anforderungen aus Benutzersicht (UML-Use-Case-Diagramme, Szenarienbeschreibungen etc. nach [14][17]) und die zugehörigen Analyse-Entwicklungsmethoden aus BOE [17] und dem Unified Process [15],
- Beschreibungstechniken für die Erstellung von Anforderungen aus Entwicklersicht (UML-Klassendiagramme, UML-Sequenzdiagramme etc. nach [14]) und die zugehörigen Analyse-Entwicklungsmethoden aus OMT [18] und dem Unified Process [15],
- Entwurfsbeschreibungstechniken (UML-Klassendiagramme, UML-Package-diagramme, UML-Zustandsdiagramme etc. nach [14][15]) und die Entwurfsentwicklungsmethoden aus OMT [18] und dem Unified Process [15],
- Kodierungstechniken- und Werkzeuge (C++, UNIX-Werkzeuge),
- Test-Techniken für die Komponenten- und System-Validation (Black-Box-Testen),

sowie bereits im SFB (weiter-)entwickelte

- Anforderungsbeschreibungstechniken (FOREST-Ansatz [1][5][7]; entwickelt in den Teilprojekten C1 und B4),
- Inspektionstechniken (Checklisten-basierte Inspektionen mit domänenspezifischen Checklisten [9]; entwickelt in Teilprojekt B1),
- Methoden und Werkzeuge zur Projekt- und Messplanung (Zielführende Projektplanung mit MVP-L [6], GQMPlanner [4]; entwickelt in Teilprojekt B1).

Diese Techniken, Verfahren und Werkzeuge wurden nach einem Wasserfall-ähnlichen Entwicklungsprozess zur Entwicklung eines kompletten Gebäudeautomationssystems eingesetzt. Als Baseline dokumentiert wurden

- das tatsächlich durchgeführte Prozessmodell [3],
- Aufwands-/Zeit-/Fehlermodelle (entsprechend des Prozessmodells) [2],
- Aufwands-/Zeit-/Fehlermodelle bezüglich einzelner Techniken, Methoden und Werkzeuge (nur rudimentär) [2].

3 Plan für Q2

Gemäß der Intention von Baselining-Entwicklungen im SFB war die Durchführung des gesamten Anwendungsentwicklungsprozesses von der OO-Analyse bis zum Systemtest vorgesehen. Ausgeschlossen blieb die Entwicklung von Kommunikations- und Betriebssystemschichten. Somit konnte das resultierende Anwendungssystem nur simulativ ausgeführt werden. Zugrundegelegt wurde ein Wasserfall-ähnlicher OO-Entwicklungsprozess, der sich am Stand der Technik orientiert und sich in ein für den

SFB standardisiertes Lebenszyklusmodell einfügt. Als wesentliche Entwicklungs- und Beschreibungstechniken wurden UML, C++, Checklisten-orientierte Anforderungs-, Entwurfs- und Kodeinspektionen sowie Black-Box-Testverfahren für die Komponenten- und System-Validation eingesetzt. Für die Planung wurden die Prozessmodellierungsansätze MVP-L und MILOS, der GQM-Messansatz sowie diese Ansätze unterstützende Planungswerkzeuge eingesetzt.

Die Aufgabenstellung lautete, ausgehend von einer im Projektbereich D (Anwendungssystem Gebäude) erstellten Problembeschreibung, ein Gebäudeautomationssystem mit Licht- und Temperaturregelung zu erstellen. Der geplante zeitliche Rahmen betrug drei Monate. Als Entwickler waren Informatikstudenten mit Software-Engineering-Kenntnissen vorgesehen, die Leitung einzelner Entwicklungsgruppen sollten wissenschaftliche Mitarbeiter übernehmen. Um die Entwicklung in dem gegebenen zeitlichen und organisatorischen Rahmen realisieren zu können, wurde eine stark arbeitsteilige Vorgehensweise angestrebt. Hierzu war die Bildung von Teams vorgesehen, die beginnend mit der Analysesphase parallel an Subsystemen mit minimaler Kopplung arbeiten. Eine entsprechend geeignete Anforderungsstruktur wurde vorgegeben. Für die Entwicklung war die Verwendung folgender Werkzeuge vorgesehen: Zur Modellierung und (Teil-)Generierung der UML-Diagramme sollte Software through Pictures (StP) von Aonix in der UML-Version verwendet werden [13]. Die Dokumentation sollte weitgehend mit dem Textverarbeitungssystem FrameMaker erfolgen. Für die automatische Generierung großer Teile der Dokumentation wurde ein eigenentwickelter Generator [8] bereitgestellt, sodass ein einfaches Zusammenspiel von StP und FrameMaker möglich war. Zur Übersetzung von Kode und zur Systemintegration waren UNIX-übliche Werkzeuge vorgesehen. Die Baselining-Entwicklung sollte auf folgenden Ergebnissen des SFB aufbauen: Neben der bereits erwähnten und im Rahmen von Prototyping-Entwicklungen (siehe Querschnittsprojekt Q1) verfeinerten Problembeschreibung war eine natürlichsprachliche, gemäß des FOREST-Ansatzes erstellte Problembeschreibung als Referenzdokument vorgesehen [12] (siehe auch Querschnittsprojekt Q4). Diese von den Teilprojekten C1 und B4 erstellte Problembeschreibung sollte als detailliertes Nachschlagewerk zusätzlich zur Problembeschreibung verwendet werden [1][5][7]. Weiterhin wurde eine Systemdokumentation eines eingeschränkten Gebäudeautomationssystems (nur Klimaregelung) aus einem früheren Entwicklungsprojekt zur Verfügung gestellt, die in erster Linie als Orientierungshilfe bzw. Richtlinie für die Dokumentation dienen sollte. Die Planung (insbesondere gemachte Sollvorgaben) basierte auf Schätzungen für Aufwand, Fehler und Kalenderzeit aus früheren Entwicklungsprojekten [10][11].

4 Durchführung von Q2

Die Baselining-Entwicklung wurde von 17 Software-Entwicklern (Informatikstudenten mit Software-Engineering-Kenntnissen) an der Universität Kaiserslautern im Jahr 1999 durchgeführt. Die Baselining-Entwicklung erfolgte im Rahmen eines dreimonatigen Software-Engineering-Praktikums, das basierend auf einer vorhergehende Software-Engineering-Vorlesung durchgeführt wurde. Die Entwickler wurden in fünf Teams zu je drei bzw. vier Personen eingeteilt. Wesentliche Aufgabe dieser Teams war die parallele Bearbeitung folgender Subsysteme: a) Temperaturregelung, b) Lichtregelung für Flure, c) Lichtregelung für Räume, d) Sensor- und Aktuatormanagement, e) Verwaltung von Systemmodi. Fünf Mitarbeiter/-innen der SFB-Teilprojekte A1 und B1 und der Arbeitsgruppe Software Engineerung betreuten die parallelen Teams unter Leitung von Prof. Rombach (Natalie Ardet, Raimund L. Feldmann, Antje von Knethen, Jürgen Münch, Stefan Vorwieger). Bei der Anforderungsanalyse sowie deren Inspektion wurde ein Mitarbeiter des SFB-Teilprojekts C1 involviert (Martin Kronenburg). Die Inspektion des Systementwurfs erfolgte unter Einbeziehung von Mitarbeitern der Teilprojekte B5 und B10 (Lothar Baum, Martin Becker, Lars Geyer und Georg Molter). Die Baselining-Entwicklung konnte entsprechend der Planung durchgeführt werden, jedoch konnten die Systemintegration und -validation nicht im Praktikum erfolgen und wurden im Anschluss von wissenschaftlichen Hilfskräften durchgeführt. Hierauf entfielen ca. 10% des Gesamtaufwandes. Die Komplettierung der Entwicklung durch wissenschaftliche Hilfskräfte war erforderlich, da der zeitliche Aufwand für Entwurfs- und Kodeinspektionen unterschätzt worden war.

5 Resultierende Modelle

Wesentliche Resultate der Baselining-Entwicklung sind ein Prozessmodell für die Entwicklung eingebetteter Software sowie eine Menge von Qualitätsmodellen:

- Das *Prozessmodell P501.V1*: Es handelt sich um eine Wasserfall-ähnliches Modell mit Iteration [3] mit folgenden Besonderheiten:
  - → Der Analyseschritt wurde zweistufig (getrennt nach Benutzer- und Entwickleranforderungen) mit Analysetechniken aus BOE [17], OMT [18] und dem Unified Process [15] durchgeführt.
  - → Der Entwurfsschritt wurde mit Entwurfstechniken aus OMT [18] und dem Unified Process [15] durchgeführt.
  - → Das Ende der Prozessschritte Analyse, Entwurf und Kodierung wurde durch einen Inspektionsschritt dokumentiert. Diese Inspektion wurde Checklisten-orientiert durchgeführt.
  - → Der Prozess legt einen Schwerpunkt auf die Beschreibung des Verhaltens des Systems. Dies spiegelt sich in der umfassenden Modellierung von Zustands-, Sequenz- und Kollaborationsdiagrammen wider.
  - → Der Prozess ist domänenspezifisch (für eingebettete Software) ausgelegt. Dies schlägt sich beispielsweise in modifizierten Notationen oder der Einbeziehung von Referenzmodellen nieder.

- Das *Aufwandsmodell A501.V1* zeigt eine (37%, 30%, 21%, 9%, 3%)-Verteilung auf die Phasen Analyse, Entwurf, Kodierung (mit integrierter Komponenten-Validation), Integration und System-Validation [9]. Besonders fällt auf, dass

→ der Aufwand für die System-Validation außergewöhnlich gering (3%) ist. In industriellen Projekten geht man i. a. von bis zu 50% aus. Der geringe Anteil kann auf die frühzeitige Entdeckung vieler Fehler bei Inspektionen zurückgeführt werden. Allerdings ist bei der Beurteilung der verwendeten Testdaten zu berücksichtigen, dass diese aus den in der Analyse erstellten Szenarien abgeleitet sind und nicht auf statistisch fundierten Benutzungsprofilen fundieren.

→ der Aufwand in den frühen Prozessschritten (Analyse und Entwurf) verhältnismäßig hoch (67%) ist. Dies ist im wesentlichen auf zwei Analyseschritte und die umfangreiche Modellierung verschiedener UML-Sichten zurückzuführen. Die Produktivität bezogen auf den Gesamtaufwand liegt jedoch mit 8 LOC/h (Lines of Code/Stunde) weit über dem Durchschnitt industrieller Projekte (5 LOC/h für Organisationen mit CMM-Level 3).

- Das *Fehlerslippagemodell FS501.V1* zeigt eine Verteilung der Fehler auf Produkte gemäß Abb. 1 [9]. Es ist nach den Produkten, in denen die Fehler auftreten, gegliedert und gibt Auskunft über den jeweiligen Zeitpunkt der Entdeckung eines Fehlers. Das Fehlerslippagemodell ist beispielhaft im Anschluss an diese Auflistung hinsichtlich seiner Bedeutung und möglicher Schlussfolgerungen interpretiert. Besonders fällt auf, dass

  → die meisten der besonders teuren Anforderungsfehler (96%) bereits frühzeitig bei der Anforderungsinspektion entdeckt wurden.

- Das *Fehlerklassenmodell FK501.V1* zeigt für die Fehlerklassen "Unvollständigkeit", "Inkonsistenz", "Falsche Information", "Überspezifikation" und "Vorwärtsreferenzen" folgende Gesamtfehlerverteilung: (13%, 11%, 8%, 3%, 4%) für das Analysedokument, (19%, 10%, 8%, 1%, 2%) für das Entwurfsdokument und (5%, 8%, 5%, 1%, 2%) für den Kode [9]. Besonders fällt auf, dass

  → in sämtlichen Fehlerklassen eine Abnahme der Fehleranzahl im Verlauf der Entwicklung festzustellen ist (mit einer Ausnahme beim Entwurf). Dies spricht für eine frühzeitige Erkennung unterschiedlichster Fehlerarten. Die hohe Zahl an Unvollständigkeiten und Inkonsistenzen bei Analyse und Entwurf lässt auf eine noch unzureichende Verfolgbarkeit zwischen den UML-Sichten schließen.

- Das *Kalenderzeitmodell K501.V1* zeigt eine (26%, 40%, 13%, 3%, 18%)-Verteilung auf die Phasen Analyse, Entwurf, Kodierung, Integration und System-Validation [9]. Die zeitliche Verteilung der Aktivitäten harmoniert weitgehend mit der Aufwandsverteilung. Abweichungen lassen sich auf die Einhaltung von Koordinationszeitpunkten zwischen Teams und damit verbundene zeitliche Verzögerungen zurückführen.

Sämtliche Qualitätsmodelle sind unter [9] abgelegt und in [2] finden sich Interpretationen der Qualitätsmodelle. Das in Abb. 1 dargestellte Fehlerslippagemodell soll hier beispielhaft interpretiert werden. Es zeigt die Verteilung der Fehler über die Problemlösungsdokumentation (diese umfasst die Analyse), die Software-System-Dokumentation (diese umfasst den Entwurf), die Software-Komponenten-Dokumentation (diese umfasst die Komponenten-Entwürfe und den Kode) sowie das ausführbare System. Die Prozesse 'System-Anforderungen bearbeiten', 'System-Entwurf bearbeiten' und 'System-Komponenten bearbeiten' umfassen jeweils eine abschließende Inspektion der erstellten Dokumente. Die meisten Fehler, die während der Analyse gemacht wurden, konnten während der Anforderungsinspektion entdeckt werden. Nur 0,46% der Analysefehler wurden erst in der Entwurfsphase und nur 0,76% bei der System-Validation gefunden. Dies spricht für eine sehr effektive Anforderungsinspektion und einen hohen Reifegrad der verwendeten Checklisten. Entwurfsfehler konnten nicht in diesem Umfang bei der Entwurfsverifikation entdeckt werden: 5,23% wurden erst in späteren Phasen entdeckt. Eine Analyse ergab, dass vor allem auf Inkonsistenzen beruhende Fehler (24%) sowie Unvollständigkeitsfehler (48%) im Entwurfsdokument vorherrschten. Dies ist auf eine unzureichende horizontale Verfolgbarkeit zwischen verschiedenen UML-Entwurfssichten zurückzuführen. Eine entsprechende Entwicklungsunterstützung durch explizite Verfolgbarkeitsbeziehungen und hierauf abgestimmte Checklisten bei der Inspektion versprechen eine signifikante Verringerung der Entwurfsfehler. Bei der Komponentenentwicklung sind ca. 20% der Fehler aufgetreten, jedoch konnten die meisten (18,28%) bei der Kodeinspektion entdeckt und daraufhin behoben werden. Da das Qualitätsmodell die Fehler entsprechend ihres Ursprungsprodukts zuordnet, sind dem ausführbaren System keine Fehler zugeordnet. Bei der System-Validation entdeckte Fehlverhalten werden im Qualitätsmodell auf Fehler in den jeweiligen Dokumentationen abgebildet.

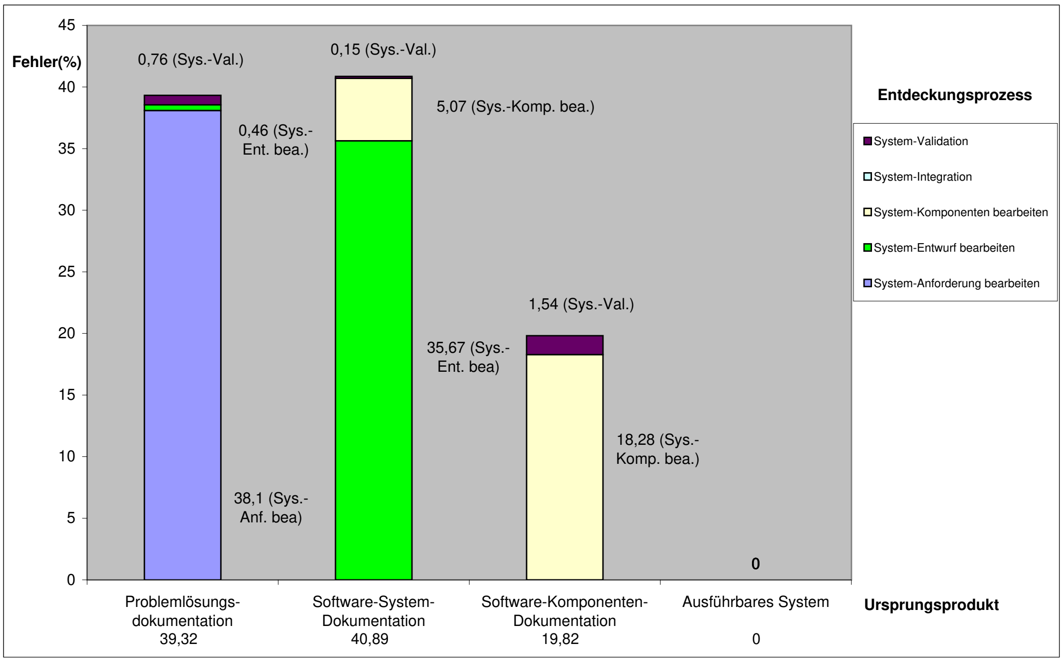

Abbildung 1: Slippagemodell für Fehler

6 Resultierende Erfahrungen

Bei der Durchführung der Baselining-Entwicklung wurden systematisch qualitative Erfahrungen in Form von Lessons Learnt dokumentiert und anschließend ausgewertet [2][9], sodass sie als Anhaltspunkte für zukünftige Verbesserungen der eingesetzten Methoden, Techniken und Werkzeuge verwendet werden können. Diese Erfahrungen sind vollständig in der SFB-Erfahrungsdatenbank abgelegt und in eine Zahl von Klassen eingeteilt, für die im Folgenden jeweils ein Beispiel angegeben ist: a) *Erfahrungen mit der Methodik:* Konflikte bei der Parallelmodellierung von Klassendiagrammen führten zur Zuordnung von Verantwortlichen für "Schnittstellenklassen"; b) *Erfahrungen mit den Entwicklungswerkzeugen:* Die werkzeuggestützte Generierung von Dokumenten führte zu einer hohen Zeitersparnis, sollte jedoch in Zukunft nur zu prozessgekoppelten Zeitpunkten erfolgen; c) *Erfahrungen mit der Planung/Management:* Der Aufwand für Inspektionen wurde unterschätzt, sodass mehr Zeit eingeplant werden sollte; d) *Erfahrungen mit Messen und Bewerten*: Die Wiederverwendung von Messerfahrungen konnte durch effektive Anpassung von existierenden GQM-Plänen erfolgen; e) *Erfahrungen mit der Wiederverwendung von Domänenwissen:* Die existierende Dokumentation reichte zur Anwendung der Steuerungsalgorithmen nicht aus und erforderte die Konsultation von Experten.

7 Offene Fragen

Die Analyse der quantitativen und qualitativen Erfahrungen führte zur Identifikation von Verbesserungspotentialen und lieferte damit Hinweise auf weitere Forschung im B- und C-Bereich. Als wesentliche offene Fragen und damit verbundene Verbesserungspotentiale wurden erkannt:

a) Wie lassen sich Unvollständigkeiten und Inkonsistenzen bei der Analyse und beim Entwurf reduzieren bzw. frühzeitiger entdecken? Der Einsatz von Werkzeugen zur Verbesserung der horizontalen Verfolgbarkeit, explizite Verfolgbarkeitsinformation zur Verbesserung der vertikalen Verfolgbarkeit sowie eine Änderung der Entwurfsinspektionen können diesbezüglich Verbesserungen bringen.

b) Wie lässt sich Domänen- und Anforderungswissen so beschreiben, dass eine weitgehend selbstständige Nutzung durch die Entwickler (ohne Expertenkonsultation) möglich ist? Der Einsatz von Mechanismen zur Generierung von Projektionen auf solches Wissen kann helfen, dieses Problem zu überwinden.

c) Wie lässt sich die Anleitung der Entwickler derart verbessern, dass die vorgegebenen Beschreibungstechniken korrekt angewendet werden, Abstimmungsprobleme bei der Erstellung verschiedener UML-Sichten minimiert werden sowie wenig erfahrene Entwickler effektiv unterstützt werden? Die Überarbeitung der Entwicklungsrichtlinien sowie die Dokumentation des Prozessmodells in Form eines Prozesshandbuchs versprechen diesbezügliche Verbesserungen.

8 Ausblick

Die Resultate der Baselining-Entwicklung sowie die identifizierten Verbesserungspotentiale sollen Änderungen oder Neuentwicklungen von Methoden, Techniken und Werkzeugen motivieren, die zunächst in den Teilprojekten isoliert erprobt und im Falle der positiven Erprobung wiederum in einer Baselining-Entwicklung in der dritten Förderungsperiode eingesetzt werden sollen. Des Weiteren ist vorgesehen, Baselining-Entwicklungen auf die Entwicklung von maßgeschneiderten Kommunikations- und Betriebssystemschichten auszuweiten (zusätzlich zur Anwendungssoftware). Mit Baselining-Experimenten wird im SFB 501 eine weitgehend realistische Entwicklungssituation angestrebt. Bei der Beurteilung der Übertragbarkeit der Resultate muss jedoch berücksichtigt werden, dass die Entwicklung mit Informatikstudenten im Hauptstudium in einem universitären Umfeld erfolgt. Die Durchführung von Baselining-Entwicklungen in industriellen Umgebungen mit professionellen, erfahrenen Entwicklern kann zu einer Steigerung der externen Validität (d. h., Übertragbarkeit) der Resultate führen. Dies soll im Kontext von Industrie-Kooperationen und Transferprojekten erfolgen.

9 Veröffentlichungen

9.1 Eigene Veröffentlichungen

[1] R. Gotzhein, M. Kronenburg, C. Peper: *Reuse in Requirements Engineering: Discovery and Application of a Real-Time Requirements Pattern*. 5th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT‘98), LNCS 1486, Springer, Lyngby, Dänemark, S. 65-74, 1998.

[2] Antje von Knethen, Raimund Feldmann, Jürgen Münch, Stefan Vorwieger: *A Baseline for the Development of Embedded Software*. Technischer Bericht Nr. 4/00, Sonderforschungsbereich 501, 67653 Kaiserslautern, 2000.

[3] Antje von Knethen, Jürgen Münch: *Entwicklung eingebetteter Software mit UML: Der Do-it-Prozess V 1.0*, Technischer Bericht Nr. 05/00, Sonderforschungsbereich 501, 67653 Kaiserslautern, 2000.

[4] Jürgen Kortgen: *Validierung der systematischen Entwicklung von GQM-Plänen.* Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2000.

[5] M. Kronenburg, C. Peper: *Definition and Instantiation of a Reference Model for Problem Specifications*. Proc. 11th International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Deutschland, S. 332-336, 1999.

[6] Jürgen Münch: *Anpassung von Vorgehensmodellen im Rahmen ingenieurmäßiger Softwarequalitätssicherung*. Im Tagungsband des 6. Workshops der Fachgruppe 5.1.1 (GI), Kaiserslautern, 19.-20. April, 1999.

[7] C. Peper, R. Gotzhein, M. Kronenburg: *A Generic Approach to the Formal Specification of Requirements*. Proc. 1st IEEE International Conference on Formal Engineering Methods (ICFEM‘97), Hiroshima, Japan, S. 252-261, 1997.

[8] Wolfgang Wagenbichler: *Erstellung einer SW-Systemdokumentation mit StP.* Projektarbeit, Universität Kaiserslautern, 67653 Kaiserslautern, 1999.

9.2 Online-Dokumentationen

[9] AG Software Engineering: *Dokumentation Querschnittsprojekt Q2*. URL: http://sep1.informatik.uni-kl.de:18000/TEAM3/WWW/BaX3/baX3.html

[10] AG Software Engineering: *Software-Entwicklungsdokumentation zum SE-1-Praktikum 1998.* URL: file:/homes/edbdemo/EDB/INHALTE/SPEZIFISCH/FALLSTUDIEN/se-1-98_contents.html

[11] AG Software Engineering: *Software-Entwicklungsdokumentation zum SE-1I-Praktikum 1998/1999.* URL: file:/homes/edbdemo/EDB/INHALTE/SPEZIFISCH/FALLSTUDIEN/SE-2-99/

[12] C. Peper, R. Gotzhein, M. Kronenburg: *Problemspezifikation nach dem FOREST-Ansatz*. URL: http://www-avenhaus.informatik.uni-kl.de/forest/EXAMPLES/SE_PRAKTIKUM/SE_Prakti

9.3 Fremdveröffentlichungen

[13] Aonix: *Software through Pictures: User Manuals*. Aonix Release 2.4 und 3.4, 1998.

[14] Grady Booch, James Rumbaugh, Ivar Jacobson: *The Unified Modeling Language User Guide*. Addison-Wesley Longman, 1999.

[15] Ivar Jacobson and Grady Booch and James Rumbaugh: *The Unified Process.* IEEE Software, Vol. 16, Nr. 3, S. 96-102, Mai 1999.

[16] Ivar Jacobson, Grady Booch, James Rumbaugh: *Unified Software Development Process*. Addison Wesley Publishing Company, 1999.

[17] Bernd Oesterreich: *Objektorientierte Softwareentwicklung mit der Unified Modeling Language*. Oldenburg-Verlag, 3., aktualisierte Auflage 1997.

[18] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, Bill Lorensen: *Object-Oriented Modeling and Design.* Prentice Hall, 1990.

Querschnittsprojekt Q3

Modellierung der Domäne Gebäudeautomation

Mitarbeiter

Dipl. Inform. Lars Geyer (B5)

Dipl. Inform. Martin Kronenburg (C1)

Dr. Rolf Merz (D2)

Dr. Bernd Schürmann (D1)

1 Aufgabenstellung

Die initiale Zielsetzung dieses Querschnittsprojekts ist es, die Erfahrungen im SFB, insbesondere in den Teilprojekten D1 und D2, bezüglich der Domäne Gebäudeautomation in einem Domänenmodell zusammenzufassen. Ein Domänenmodell beschreibt das Wissen über ein bestimmtes Anwendungsfeld im Problemraum, d.h. es werden alle relevanten Anforderungen an mögliche Systeme in einer Domäne, sowie alle für das Verständnis des Problems notwendigen Informationen in einem Modell zusammengefasst.

Durch ein Domänenmodell werden im Wesentlichen zwei Ziele erreicht. Zum einen dient die Modellierung dazu, eine Domäne abzugrenzen. Ein Domänenmodell definiert die mögliche Anforderungsbandbreite, die bei der Erstellung eines Systems berücksichtigt werden muss. Zum anderen ist eine wesentliche Aufgabe eines Domänenmodells die Unterscheidung zwischen Gemeinsamkeiten und Variabilitäten. Gemeinsamkeiten sind Anforderungen, die an alle Systeme einer Domäne gestellt werden. Im Gegensatz dazu stehen variable Anforderungen, die nur in bestimmten Systemen einer Domäne eine Rolle spielen. Durch die explizite Modellierung von Gemeinsamkeiten erlaubt ein Domänenmodell eine Abschätzung bezüglich des Wiederverwendungspotentials in einer Domäne.

Als Grundlage dienten dem Querschnittsprojekt die von den Teilprojekten D1 und D2 während der zurückliegenden Förderungsperioden entwickelten Problembeschreibungen. Zudem stand durch die Teilnahme von Mitarbeitern dieser Teilprojekte Expertenwissen unmittelbar zur Verfügung.

Als initiale Vorgehensmethode wurde die *Commonality Analysis* aus dem *FAST*-Ansatz für *Product-Line Software Engineering* [7] verwendet, die in Abschnitt 2.2 vorgestellt wird. Die mit dieser Methode erzielten Ergebnisse waren jedoch nicht zufrieden stellend, so dass im Rahmen des Querschnittsprojekts die Methode weiterentwickelt wurde. Die resultierende Methode und das daraus abgeleitete Strukturierungskonzept für Domänenmodelle werden in den Abschnitten 2.2 und 2.3 beschrieben. Eine genauere Beschreibung der erzielten Ergebnisse ist in SFB-Bericht [1] zu finden.

2 Durchgeführte Arbeiten und Ergebnisse

2.1 Begriffserklärungen

Zunächst war es erforderlich, eine einheitliche Begriffswelt festzulegen. Dabei konnte auf die in [3] eingeführte Terminologie zurückgegriffen werden. Danach ist ein *System* ein Teil der realen Welt, der aus bestimmten Gründen als eine Gesamtheit angesehen wird. In einem System sind verschiedene Phänomene vereint und miteinander verbunden. Ein *Phänomen* ist ein Aspekt der realen Welt, der wesentlich für ein betrachtetes System ist. Phänomene sind zum Beispiel Zustände eines Systems wie die Temperatur in einem Raum, Ereignisse wie das Einschalten einer Lampe, Objekte wie Sensoren oder aber auch Personen wie zum Beispiel der Hausmeister eines Gebäudes. Um über solche Phänomene und damit über ein System reden zu können, werden Bezeichner eingeführt. Ein *Bezeichner*, zum Beispiel ein Wort in natürlicher Sprache oder ein Prädikatssymbol in einer Logik, ist also ein Repräsentant eines bestimmten Phänomens eines Systems. Um Mehrdeutigkeiten zu vermeiden, fordern wir stets eine Eins-zu-eins-Beziehung zwischen Phänomenen und Bezeichnern. Außerdem verlangen wir, dass es zu jedem Bezeichner eine Erläuterung gibt, die beschreibt, welches Phänomen durch einen Bezeichner repräsentiert wird. In [6] wird die Wichtigkeit solcher Erläuterungen betont.

Eine *Aussage* drückt einen Zusammenhang zwischen einigen Phänomenen aus. Jede Aussage besteht im Wesentlichen aus den Bezeichnern, die die betrachteten Phänomene repräsentieren. Der Satz „Wenn ein Raum belegt ist, muss das Licht an sein“ oder die Formel „raumBelegt -> lichtAn“ sind Beispiele für Aussagen. Eine *Beschreibung* besteht aus einer Menge von Bezeichnern versehen mit Erläuterungen sowie einer Menge von Aussagen, in denen ausschließlich die angegebenen Bezeichner vorkommen. Zu beachten ist, dass die Angabe der Phänomene in den Erläuterungen der Bezeichner auch festlegt, aus welcher Sicht und auf welchem Abstraktionsgrad ein System betrachtet wird.

Eine *Domäne* ist schließlich eine Menge von Systemen, die etwas gemeinsam haben, sowie das Wissen über diese Systeme. Ein *Domänenmodell* ist eine Beschreibung einer Domäne. Zwischen einzelnen Domänen kann es eine große Anzahl von Beziehungen geben, zum Beispiel ist die Domäne Gebäudeautomation eine *Teildomäne* der Domäne *reaktive Systeme*. Weitere Relationen entstehen, wenn nur ein Ausschnitt oder eine andere Abstraktionsebene einer Domäne beschrieben wird.

2.2 Arbeitsbericht

Ausgangspunkt für die Erstellung des Domänenmodells war die Methode der *Commonality Analysis*, die im Rahmen des *FAST*-Ansatzes zur Domänenanalyse verwendet wird. Die Vorgehensweise in diesem Ansatz basiert auf einer Gesprächsrunde, in der sich Domänenexperten treffen, um über die Aspekte der betrachteten Domäne zu diskutieren. Ein Moderator führt durch die Diskussion. Zusätzlich sorgt der Moderator dafür, dass die Ergebnisse unmittelbar in einem Dokument erfasst werden, so dass die Mitglieder jederzeit Zugriff auf den aktuellen Stand der Diskussion haben.

Ein Dokument nach dem *FAST*-Ansatz ist in mehrere Kapitel unterteilt. Es existiert jeweils ein Abschnitt für die Gemeinsamkeiten der betrachteten Systeme, deren Variabilitäten, die jeweiligen Bandbreiten der Variabilitäten sowie ein *Dictionary* der relevanten Begriffe. Dies wird ergänzt durch einen speziellen Abschnitt für offene Fragen. Die Phänomene der Domäne werden während der Analyse an Hand ihrer Bedeutung als Gemeinsamkeit oder als Variabilität klassifiziert und dementsprechend in den relevanten Abschnitt eingefügt, wobei die Abschnitte im Wesentlichen aus einer Liste der betrachteten Phänomene bestehen. Für variable Phänomene wird zusätzlich noch die Bandbreite festgelegt, in der sie in der Domäne auftreten. Treten während eines Treffens offene Fragen auf, so werden diese in dem speziellen Abschnitt dokumentiert und als Hausaufgabe an Mitglieder der Gruppe verteilt. Diese sammeln die benötigten Informationen, so dass im darauf folgenden Treffen eine Entscheidung bezüglich dieser offenen Fragen getroffen werden kann.

Um die Abhängigkeiten zwischen den einzelnen Phänomenen zu dokumentieren, wird in den einzelnen Beschreibungen auf verwandte Phänomene verwiesen. Ebenso wird die Entscheidungsfindung bei offenen Fragen dokumentiert, indem eine Fragestellung mit den die Entscheidungen betreffenden Phänomenen verknüpft wird.

In einem ersten Schritt wurde im Querschnittsprojekt dieser Ansatz verwendet, um ein Modell der Domäne Gebäudeautomation zu erstellen. Im Laufe der Zeit wurde die Vorgehensweise jedoch an die Gegebenheiten der Gruppe angepasst. So war zum Beispiel auf Grund der Gruppenstärke von vier Personen kein Moderator zur Leitung der Diskussionen nötig. Alle vier Gruppenmitglieder waren gleichberechtigte Teilnehmer an den Diskussionen. Des Weiteren wurde das Dokument nicht unmittelbar während der Treffen erweitert. Statt dessen wurden Teile des Dokumentes von einzelnen Mitgliedern der Gruppe als Hausaufgabe zwischen den Treffen erstellt. In den Treffen wurden die neuen Teile dann systematisch durchgegangen und in der gesamten Gruppe diskutiert.

Eine wesentliche Änderung zur *Commonality Analysis* stellt die Form dar, in der das Domänenmodell dokumentiert wird. Bei der konkreten Anwendung zeigte sich, dass die starre Trennung zwischen Gemeinsamkeiten und Variabilitäten erzwungen erscheint. Des Weiteren schien auch die listenartige Aufzählung der Phänomene, wie dies in der *Commonality Analysis* normalerweise vollzogen wird, nicht adäquat. Daher entwickelte die Gruppe eine Technik zur strukturierten, objektorientierten Formulierung eines Domänenmodells. Da dieses Strukturierungskonzept ein wesentliches Ergebnis des Querschnittsprojektes darstellt, wird es im nächsten Abschnitt explizit beschrieben.

2.3 Das Strukturierungskonzept

Der strukturierte Aufbau des Domänenmodells basiert im Wesentlichen auf den Strukturierungskonzepten, die in der Kooperation der Teilprojekte B4 und C1, siehe auch Querschnittsprojekt Q4, zur objektorientierten Strukturierung von Problemspezifikationen entwickelt worden sind, vgl. [3], [4]. Um ein Domänenmodell, das eine Menge *großer* Systeme repräsentiert, adäquat darstellen zu können, verwenden wir die Konzepte Modularisierung, Vererbung, Aggregation und Parametrisierung.

Ein Domänenmodell wird als eine Folge von *Beschreibungsklassen* dargestellt. Eine Beschreibungsklasse enthält ihrerseits sowohl Informationen bezüglich der Vererbungs- und Aggregationsbeziehung zu anderen Beschreibungsklassen als auch eine Liste von Bezeichnern, zusammen mit Erklärungen hinsichtlich der repräsentierten Phänomene, sowie eine Liste von indikativen und optativen Eigenschaften. Durch

Beschreibungsklassen ist es möglich, einzelne Teile der Systeme, die zu einer Domäne gehören, getrennt und damit übersichtlicher zu beschreiben. Abbildung 1 zeigt einen Ausschnitt aus der graphischen Repräsentation der Aggregations- und Vererbungshierarchie des erarbeiteten Domänenmodells.

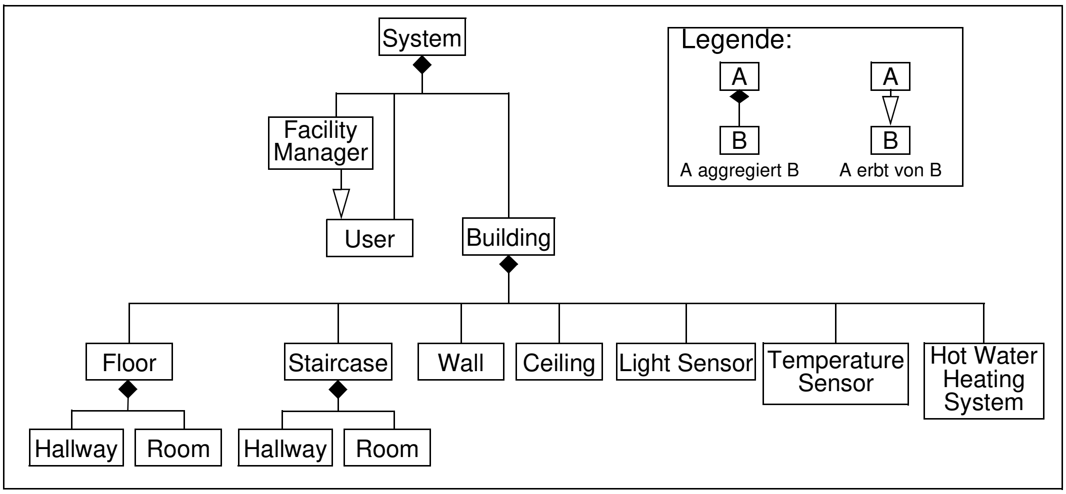

Abbildung 1: Ausschnitt aus der Aggregations- und Vererbungshierarchie

Eine Parametrisierung einer Beschreibungsklasse ist zum einen im Rahmen der Aggregation mit Hilfe von *Anzahlparametern* möglich. Dadurch kann die genaue Anzahl der aggregierten Objekte, zum Beispiel die der Räume auf einem Flur, offen gelassen werden. Zum anderen können im Eigenschaftsteil durch bedingte Aussagen weitere Variabilitäten ausgedrückt werden. Abbildung 2 zeigt einen Ausschnitt der Beschreibungsklasse Door.

2.4 Anwendung des Domänenmodells im SFB

Instanziierung des Domänenmodells

Nachdem ein initiales Domänenmodell vorliegt, muss untersucht werden, inwieweit das Modell als Eingabe für den SFB-Entwicklungsprozess herangezogen werden kann. Vor Durchführung eines entsprechenden ausführlichen Experiments sollte das Domänenmodell mit der im Querschnittsprojekt Q1 verwendeten Problembeschreibung verglichen werden. Ist bei der Instanziierung des Domänenmodells eine einfache Abbildung auf die verwendete Problembeschreibung gegeben, kann es in Zukunft im SFB eingesetzt werden, ohne dass größere Änderungen am Entwicklungsprozess notwendig wären.

Der aktuelle Stand des Domänenmodells und die im Querschnittsprojekt Q3 verwendete Problembeschreibung sind in den SFB-Berichten [1] bzw. [2] enthalten. Ein genauerer Vergleich der beiden Dokumente zeigt Unterschiede in drei wesentlichen Aspekten:

- Bei der Erstellung des Domänenmodells wurde darauf geachtet, dass dieses nur Anforderungen enthält. Die in Q3 verwendete Problembeschreibung enthält dagegen bereits Lösungsstrategien, wie sie für die Domäne typisch sind.

Class:	**Door**	(category: building)
Description:	A door is an opening in a wall which users can use for moving between different building units or the environment. It does not store heat, transmits heat, and if it is not closed it also transmits light.	
Derived Cl.:		
Inherited from:	*ShapeComponent*	
Parts:	(0..1) window: *Window* (0..1) closedSensor: *DoorClosedSensor* (0..1) lockedSensor: *DoorLockedSensor.*	
Items:	• *closed* : BOOLEAN This item represents whether a door is closed or not in the real world. It has not to be confused with the value of a possibly installed *closed sensor* measuring the value of *closed*. *closed* = 1 means that a door is closed, *closed* = 0 means that a door is not closed. • *locked* : BOOLEAN This item represents whether a door is locked or not in the real world. It has not to be confused with the value of a possibly installed *locked sensor* measuring the value of *locked*. *locked* = 1 means that a door is locked, *locked* = 0 means that a door is not locked.	
Properties:	indicative: • A door transmits heat, i.e. *heatTransmission > 0.* • A door does not store heat, i.e. *heatCapacity = 0.* • A door is pervious for a user, i.e *userPervious = 1.* • If a door is *closed* it does not transmit light, i.e. *lightTransmission = 0.* • If a door is *not closed* it transmits light, i.e. *lightTransmission > 0.* • *A door* can only be opened or closed manually, i.e. no automatic doors are considered in the domain.	

Abbildung 2: Ausschnitt aus der Beschreibungsklasse Door. Einige der benutzten Bezeichner (z.B. heatTransmission) werden von der Klasse ShapeComponent geerbt.

Konsequenz: Die Mitglieder des Querschnittsprojekts erachten die frühzeitige Bereitstellung allgemeiner Lösungsstrategien als wichtig und typisch für die Domäne. Auch wenn diese Lösungsansätze nicht verworfen werden sollten, sollten sie doch getrennt von den Anforderungen formuliert werden. Sie sollten Teil einer erweiterten Domänenmodellierung sein. Eine explizite Trennung von Anforderungen und Lösungsansätzen wird als wesentlich erachtet. Diese Thematik soll in der nächsten Förderungsperiode durch das Querschnittsprojekt vertieft betrachtet werden.

- Fast alle Anforderungen der Problembeschreibung, die nicht durch das Domänenmodell abgedeckt sind, sind nicht-funktionaler Art. Diese wurden bei der bisherigen Erstellung des Domänenmodells unberücksichtigt gelassen.
  Konsequenz: Die Gruppe ist der Meinung, dass es noch weitergehender Untersuchungen bedarf, in welcher Form nicht-funktionale Anforderungen spezifiziert werden sollten. Eine Trennung von funktionalen und nicht-funktionalen Anforderungen erscheint sinnvoll.

Insgesamt kommen die Mitglieder dieses Querschnittsprojekts zu dem Schluss, dass eine relativ einfache Integration des Domänenmodells in den SFB-Entwicklungsprozess gegeben zu sein scheint. Eine endgültige Aussage ist allerdings erst nach Durchführung entsprechender Experimente möglich. Eine Fortführung des Querschnittsprojekts in der kommenden Förderungsperiode erscheint deshalb notwendig.

Teilprojekt B10

Für das Teilprojekt B10 gibt es zwei Ansatzmöglichkeiten, die Ergebnisse des Querschnittsprojekts in die Arbeiten einfließen zu lassen. Zum einen bildet ein Domänenmodell die Basis für ein generisches Anforderungsmodell, welches Wiederverwendung bereits auf der Ebene der Anforderungsanalyse ermöglicht. Dazu sind die spezifizierten Anforderungen in einer für diese Analyse geeigneten Technik zu modellieren, d.h. die Anforderungen des Domänenmodells müssen gegebenenfalls in eine für den Zweck der Analyse geeignetere Technik umformuliert werden.

Zum anderen dient ein Domänenmodell als Basis für die Erstellung eines vorkonfektionierten Systems. Das Domänenmodell spezifiziert die Anforderungen, die beim Einsatz eines vorkonfektionierten Systems bei der Anwendungsentwicklung zu erwarten sind. Im vorkonfektionierten System muss daher sichergestellt werden, dass die verwendete Architektur die Flexibilität des Domänenmodells unterstützt, um effektiv während einer Anwendungsentwicklung als Grundgerüst einsetzbar zu sein. Zusätzlich bieten gerade die als gemeinsam klassifizierten Anforderungen genug Wiederverwendungspotential, um sinnvoll im Voraus als wiederverwendbare Artefakte implementiert zu werden.

Teilprojekt B11

Teilprojekt B11 will verstärkt nicht-funktionale Eigenschaften betrachten. Eine wichtige Aufgabe von B11 wird daher die Erweiterung des bisher vorliegenden Domänenmodells um nicht-funktionale Anforderungen sein. Diese wurden in diesem Querschnittsprojekt bisher nicht betrachtet. Für die Bearbeitung nicht-funktionaler Eigenschaften ist domänenspezifisches Expertenwissen notwendig, weshalb eine Erweiterung des Domänenmodells um solches Expertenwissen gewünscht wird (vgl. auch Teilprojekt D1). Eine weitere wichtige Aufgabe von B11 wird die Betrachtung der Entwurfsphase sein, in der die meisten nicht-funktionalen Eigenschaften bearbeitet werden. In diesem Zusammenhang soll gemeinsam mit C1 die Erweiterung des Domänenmodells in Richtung Entwurfsphase untersucht werden.

Teilprojekt C1

Aus der Sicht von Teilprojekt C1 ergeben sich vor allem drei Fragen, die im Zusammenhang mit dem bisher erstellten Domänenmodell relevant sind. Erstens: Inwieweit können neben den bisher eingesetzten Strukturierungskonzepten auch die im Teilprojekt C1 entwickelten und verwendeten formalen Beschreibungstechniken eingesetzt werden, um das Domänenmodell auf eine präzisere semantische Grundlage zu stellen? Zweitens: Welche Erweiterungen müssen an dem bisher entwickelten Referenzmodell für Problemspezifikationen vorgenommen werden, um auch solche Domänenmodelle zu erfassen? Hier sind sicherlich weitere Klassifikationsmöglichkeiten sowohl der Bezeichner als auch der Eigenschaften erforderlich. Drittens: Wie kann ein solches Domänenmodell für den Bereich der Anforderungen auf den Bereich der Entwürfe fortgesetzt werden? Insbesondere der Aspekt der Verfolgbarkeit ist hier von Interesse.

Teilprojekt D1

Auch für Teilprojekt D1 stellen sich drei Fragen im Zusammenhang mit dem Domänenmodell. Da das Domänenmodell die Bandbreite aller möglichen Systeme der Domäne definiert, kann aus ihm auch die Bandbreite der für das Testen und Prototyping benötigten Simulatoren und Testfelder abgeleitet werden. Ein weiterer Aspekt ist die Erweiterung des Domänenmodells um Expertenwissen. Dieses sollte im Wesentlichen in Form domänenspezifischer, generischer Lösungsansätze modelliert werden. Schließlich beschäftigt sich D1 mit der Entwicklung einer (Entwurfs-) Systematik, die vom Entwicklungsprozess beeinflusst wird. In diesem Zusammenhang ist zu untersuchen, wie der vorliegende und im Querschnittsprojekt Q1 erprobte SFB-Entwicklungsprozess zu erweitern ist, damit eine spezielle Problembeschreibung aus dem Domänenmodell in geeigneter Weise abgeleitet werden kann.

Teilprojekt D2

Eine interessante Anwendungsmöglichkeit des generischen Domänenmodells eröffnet sich aus der Sicht des Teilprojektes D2. Der Entwurf konkreter, instantiierter Steuerungs- und Regelungsalgorithmen benötigt detailliertes Wissen zum dynamischen Verhalten des zu beeinflussenden physikalisch, technischen Prozesses, der sogenannten Strecke. Dies leistet ein mathematisches Modell. Neben der Analyse der Wirkungszusammenhänge und Festlegung der Regelkreise, der Auswahl geeigneter Algorithmen und Bestimmung ihrer optimalen Parameter, ermöglicht es simulative Tests der prototypischen Algorithmen. Zusammen mit der Regelungsaufgabe entspricht es einer formalen Spezifikation regelungstechnisch relevanter Aspekte der Anforderungsbeschreibung [5]. Die Schwierigkeit ist seine Erzeugung, basierend auf informell spezifizierten Anforderungsdokumenten und Gebäudebeschreibungen. Im Teilprojekt D2 wurde eine generische Entwurfsmethode mathematischer Modelle zur Simulation thermischen Gebäudeverhaltens entwickelt. Die Methode beruht auf der Nutzung einer Modellbibliothek, bestehend aus modularen, objektorientiert entworfenen Komponentenmodellen mit klaren Schnittstellendefinitionen und Regeln zu ihrer Nutzung. Durch Selektion, Adaption und Komposition können instantiierte mathematische Simulationsmodelle für die konkrete Anwendung generiert werden. Die Struktur der objektorientierten, mathematischen Modellkomponenten spiegelt exakt die Vererbungs- und Aggregationshierarchie des hier entwickelten Domänenmodells wider. Eine Erweiterung des Domänenmodells um mathematische Beschreibungsformen und Modellkomponenten bietet sich an. Bei der Instantiierung eines konkreten Anforderungsdokumentes kann so simultan das mathematische Simulationsmodell als Vorlage für den Regelungstechniker generiert werden, um die Anforderungsbeschreibung bezüglich der für ihn notwendigen technisch, physikalischen Detaillierung zu erweitern.

3 Eigene Veröffentlichungen im Berichtszeitraum

SFB-Berichte

[1] L. Geyer, M. Kronenburg, R. Merz, B. Schürmann: *Domain Modeling in the SFB 501*, SFB 501 Bericht 07/00, Fachbereich Informatik, Universität Kaiserslautern, 2000

[2] R. Feldmann, J. Münch, S. Queins, S. Vorwieger, G. Zimmermann: *Baselining a Domain-Specific Software Development Process*, SFB 501 Bericht 02/99, Fachbereich Informatik, Universität Kaiserslautern, 1999

Tagungsbeiträge

[3] M. Kronenburg, C. Peper: *Definition and Instantiation of a Reference Model for Problem Specifications*, 11th International Conference on Software Engineering and Knowledge Engineering (SEKE‘99), Kaiserslautern, S. 332-336, 1999.

Zeitschriften / Monographien

[4] M. Kronenburg, C. Peper: *Application of the FOREST Approach to the Light Control Case Study*, erscheint in *Special Issue on RequirementsEngineering: The Light Control Case Study*, Journal of Universal Computer Science, 2000.

[5] R. Merz, L. Litz: *Objektorientierte mathematische Modellierung: Generische Methoden bei komplexen dynamischen Systemen*, Informatik Spektrum 2(2000), Springer Verlag Heidelberg, S. 90-99

4 Fremdveröffentlichungen

[6] M. Jackson: *A Discipline of Description (Keynote Talk)*, Requirements Engineering 3(2), S. 73-78, 1998.

[7] D. Weiss: *Software Product-Line Engineering: A Family-Based Software Development Process*, Addison Wesley, 1999

Querschnittsprojekt Q4

Anforderungsanalyse eines Gebäudeautomationssystems

Leiter

Prof. Dr. J. Avenhaus (C1)

Prof. Dr. R. Gotzhein (B4)

Mitarbeiter

Dipl. Inform. M. Kronenburg (C1)

Dipl. Inform. C. Peper (B4)

1 Aufgabenstellung

Aufbauend auf den Ergebnissen aus der Kooperation der Teilprojekte B4 und C1 in der ersten Förderperiode, die in den FOREST-Ansatz eingeflossen sind [12], sollten im Rahmen dieses Querschnittsprojekts mehrere Fallstudien durchgeführt werden. Dabei wurde stets eine gegebene natürlich-sprachliche Problembeschreibung aus dem Anwendungsfeld Gebäudeautomation in eine formale, objektorientierte Problemspezifikation überführt. Die Ziele dabei waren zum einen, allgemein die Eignung des entwickelten Ansatzes für dieses Anwendungsfeld zu überprüfen, und zum anderen, speziell die Möglichkeiten der Wiederverwendung von Teilen einer Problemspezifikation zu untersuchen.

2 Durchgeführte Arbeiten und Ergebnisse

Insgesamt wurden im Rahmen dieses Querschnittsprojekts drei Fallstudien aus dem Anwendungsfeld Gebäudeautomation bearbeitet. Erstens wurde als Unterstützung für die Arbeiten im Querschnittsprojekt Q2 eine Problemspezifikation erstellt, die den Studenten zur Verfügung gestellt wurde [3]. Zweitens nahmen die Mitglieder des Querschnittsprojekts an dem Dagstuhl-Seminar *Requirements Capture, Documentation, and Validation* teil und stellten dort Ausschnitte ihrer Problemspezifikation als vernetztes Geflecht von html-Dokumenten vor [4], [8], [13]. Drittens beteiligten sich die Mitarbeiter des Querschnittsprojekts an der internationalen Fallstudie *The Light Control Case Study* des Journal of Universal Computer Science (J.UCS) [5], [14].

Betrachtet wurde in diesen Fallstudien jeweils ein Stockwerk eines Universitätsgebäudes. Im Falle des Querschnittsprojekts Q2 wurden die Aspekte Licht und Temperatur betrachtet, in den beiden anderen Fallstudien jeweils der Aspekt Licht. Dass es sich hierbei stets um *große* Systeme gehandelt hat, soll anhand einiger Zahlen aus der J.UCS-Fallstudie belegt werden (die beiden anderen Fallstudien bewegen sich in derselben Größenordnung): In der Problemspezifikation zu der J.UCS-Fallstudie werden implizit ungefähr 10000 Phänomene betrachtet und implizit über 6000 Eigenschaften spezifiziert. Aufgrund der im FOREST-Ansatz verwendeten Strukturierungskonzepte mussten aber nur rund 160 Phänomene und rund 170 Eigenschaften explizit angegeben werden. *Explizit* bedeutet dabei zum Beispiel, dass in der Spezifikation der Beschreibungsklasse Room ein Bewegungsmelder nur einmal *explizit* angegeben wird. Durch die Aggregation von Instanzen der Beschreibungsklasse Room in der Beschreibungslasse Floor wird jedoch implizit für jeden aggregierten Raum ein solcher Bewegungsmelder *implizit* spezifiziert.

Die bei diesen Fallstudien gesammelten Erfahrungen geben berechtigten Anlass zu der Aussage, dass der FOREST-Ansatz geeignet ist, funktionale und zeitbehaftete Eigenschaften großer Systeme im Anwendungsfeld Gebäudeautomation präzise und verständlich zu erfassen. Dies wird im Folgenden genauer begründet.

Die von uns in der ersten Förderperiode mit maßgeschneiderten Operatoren angereicherte Temporallogik hat sich als ausdruckstark genug erwiesen, die erforderlichen funktionalen und zeitbehafteten Eigenschaften präzise zu spezifizieren. Wesentlich für das Verständnis einer Problemspezifikation ist die an mehreren Stellen vorgenommene Integration natürlicher Sprache. Vor allem die aus den temporallogischen Spezifikationen abgeleiteten natürlich-sprachlichen Übersetzungen und die für jede Signaturgröße, zum Beispiel Funktions- und Prädikatssymbole, verpflichtend vorgegebene Angabe einer Erläuterung des Phänomens, das durch eine Signaturgröße repräsentiert wird, tragen sehr zur Erhöhung der Verständlichkeit bei. Auf die Bedeutung solcher Erläuterungen wird auch in [15] hingewiesen.

Die in die Temporallogik integrierten Strukturierungskonzepte Modularisierung, Aggregation, Spezialisierung (in Verbindung mit Vererbung) und Parametrisierung haben sich als ausreichend mächtig erwiesen, Problemspezifikationen *großer* Systeme kompakt und damit übersichtlich darzustellen. Eine theoretische Fundierung dieser Konzepte, vor allem auch ihrer Semantik wird in [2] gegeben.
Jedoch hätten diese Fallstudien trotz der Strukturierungskonzepte nicht in dem Umfang durchgeführt werden können, wenn nicht auch ein entsprechendes Werkzeug [1] zur Verfügung gestanden hätte. Dieses Werkzeug erlaubt eine komfortable Eingabe und Verwaltung von Problemspezifikationen, garantiert deren syntaktische Konsistenz, unterstützt - teilweise automatisch - verschiedene Verfolgbarkeitsrelationen und erzeugt ansprechende Ausgaben, vor allem in Form einer Menge durch Hyperlinks verbundener html-Dokumente, siehe zum Beispiel [4], [5].

In den durchgeführten Fallstudien war, neben den durch die Strukturierungskonzepte gegebenen Wiederverwendungsmöglichkeiten innerhalb einer Problemspezifikation, eine Fallstudien übergreifende Wiederverwendung von Teilen einer Problemspezifikation auf drei Ebenen möglich:

- Auf der Ebene der Eigenschaften konnte eine große Anzahl von Eigenschaften durch Instanziierungen von *Requirement Patterns* spezifiziert werden.
- Des Weiteren war es möglich, einzelne Beschreibungsklassen, leicht abgeändert, in einer anderen Fallstudie zu verwenden.
- Schließlich hat sich gezeigt, dass die Zusammenfassung von mehreren Beschreibungsklassen zu Gruppen, zum Beispiel zur Gruppe aller Sensoren, hilfreich ist und in allen Fallstudien sehr ähnlich vorgenommen werden konnte.

3 Eigene Veröffentlichungen im Berichtszeitraum

Diplomarbeiten

[1] T. Schmidt-Samoa: *FOREST - Entwurf und Implementierung einer Umgebung zur Erstellung formaler System-Anforderungs-Beschreibungen*; Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1999.

Dissertationen

[2] M. Kronenburg: *Ein Ansatz zur Erstellung präziser, verständlicher Problemspezifikationen großer reaktiver Systeme*; Dissertation, in Vorbereitung, 2000.

WWW-Dokumente

[3] http://www-avenhaus.informatik.uni-kl.de/forest/EXAMPLES/SE_PRAKTIKUM/SE_PraktikumEnglishStart.html

[4] http://www-avenhaus.informatik.uni-kl.de/forest/EXAMPLES/DAGSTUHL/DagstuhlEnglishStart.html

[5] http://www-avenhaus.informatik.uni-kl.de/forest/EXAMPLES/JUCS/JUCSStart.html

SFB-Berichte

[6] R. Gotzhein, M. Kronenburg, C. Peper: *Reuse in Requirements Engineering: Discovery and Application of a Real-Time Requirement Pattern*; Technischer Bericht SFB 501 08/98, Fachbereich Informatik, Universität Kaiserslautern, 1998.

Vorträge

[7] R. Gotzhein: *Pattern-based System Development*; Kolloquium, Ecole Polytechnique Federale a Lausanne, 9. Juli 1998.

[8] M. Kronenburg: *xforest: A Tool for the Creation of WWW-based Problem Specifications;* Dagstuhl-Seminar Requirements Capture, Documentation, and Validation, 13.-18. Juni 1999.

[9] C. Peper: *Wiederverwendung bei der Anforderungsspezifikation: Der FOREST-Ansatz*, Workshop “Software-Reuse” des SFB 501, Kaiserslautern, 22.-23. Februar 1999.

Tagungsbeiträge
(Der Vortragende ist unterstrichen.)

[10] C. Peper, R. Gotzhein, M. Kronenburg: *A Generic Approach to the Formal Specification of Requirements*; Proc. 1st IEEE International Conference on Formal Engineering Methods (ICFEM`97), Hiroshima, Japan, S. 252-261, 1997.

[11] R. Gotzhein, M. Kronenburg, C. Peper: *Reuse in Requirements Engineering: Discovery and Application of a Real-Time Requirements Pattern*; 5th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT`98), LNCS 1486, Springer, Lyngby, Dänemark, S. 65-74, 1998.

[12] M. Kronenburg, C. Peper: *Definition and Instantiation of a Reference Model for Problem Specifications*; 11th International Conference on Software Engineering and Knowledge Engineering (SEKE‘99), Kaiserslautern, Deutschland, S. 332-336, 1999.

[13] R. Gotzhein, M. Kronenburg, C. Peper: *Pattern-based Requirements Capture Applied: The SFB 501 Case Study*; in E. Börger, B. Hörger, D. Parnas, D. Rombach, Requirements Capture, Documentation, and Validation, Dagstuhl-Seminar-Report 242, 1999.

Zeitschriften / Monographien

[14] M. Kronenburg, C. Peper: *Application of the FOREST Approach to the Light Control Case Study;* erscheint in Special Issue on “Requirements Engineering: The Light Control Case Study”, Journal of Universal Computer Science (J.UCS), Springer, 2000.

4 Fremdveröffentlichungen

[15] M. Jackson: *A Discipline of Description (Keynote Talk)*; Requirements Engineering 3(2), S. 73-78, 1998.