# 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

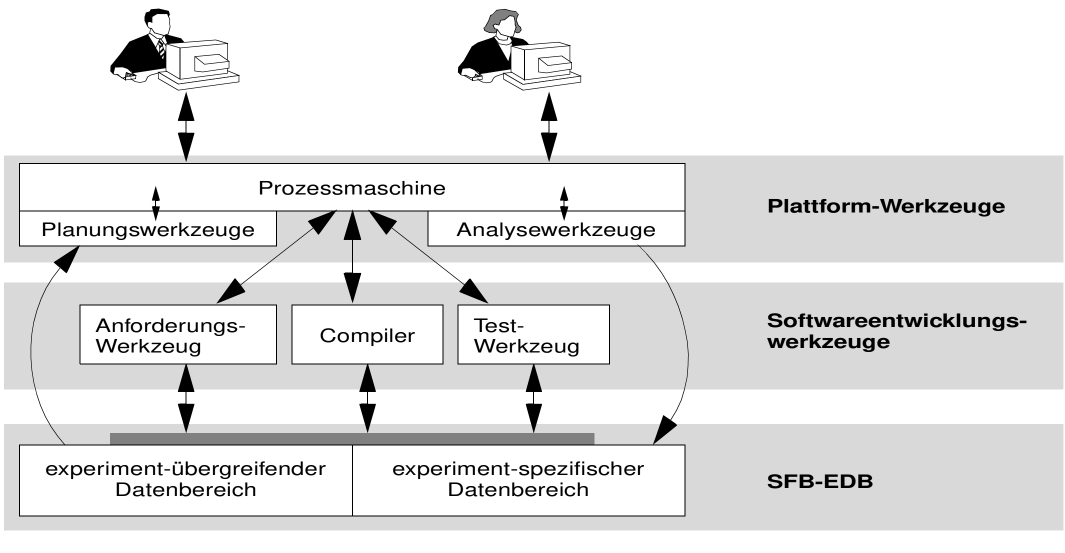


**Abbildung 1:** Schnittstellen zwischen den Technologie-Schwerpunkten des SE-Labors

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 experiment-übergreifenden 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

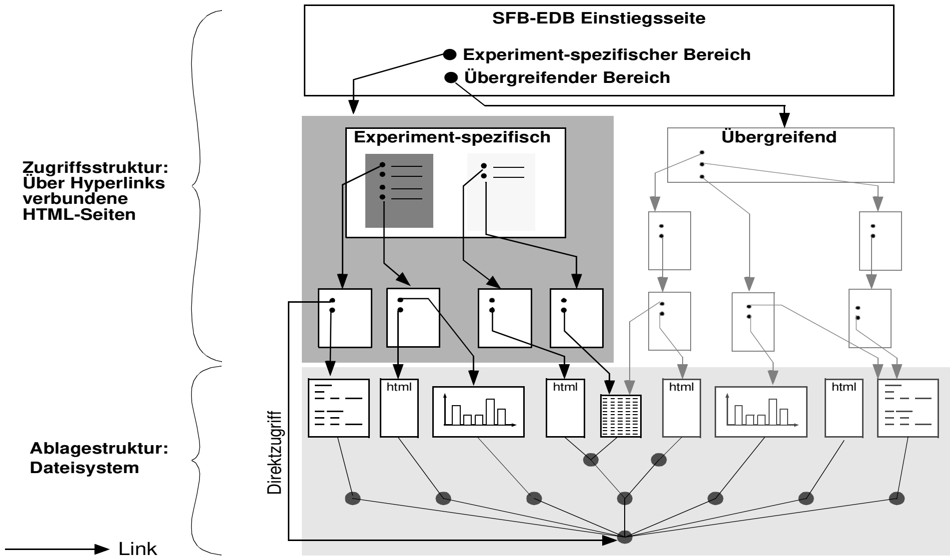


**Abbildung 2:** Ablage- und Zugriffstruktur des web-basierten SFB-EDB Prototyps

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

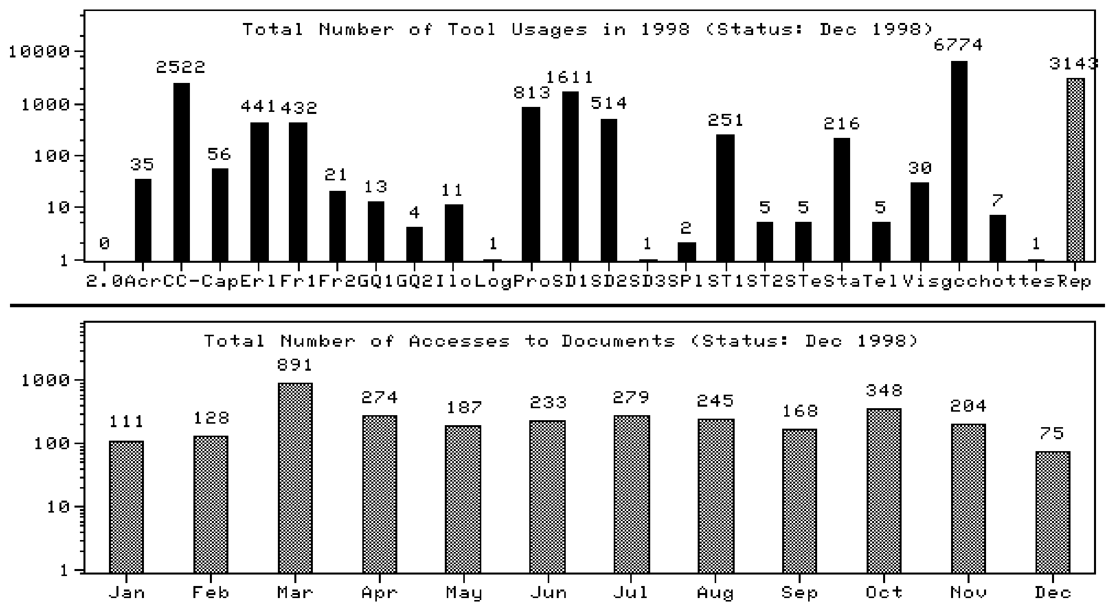


**Abbildung 3:** Zugriffe auf Werkzeuge des SE-Labors (inkl. SFB-EDB Prototyp)

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 | Prof. Dr. Dieter Rombach |
|---|---|
| geb. 21.6.1938 | geb. 6.6.1953 |
| Universität Kaiserslautern | Universität Kaiserslautern |
| Postfach 3049 | Postfach 3049 |
| 67653 Kaiserslautern | 67653 Kaiserslautern |
| Tel.: 0631/205-2800 | Tel.: 0631/205-2895 |
| Fax: 0631/205-3557 | Fax: 0631/205-3331 |
| richter@informatik.uni-kl.de | 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.

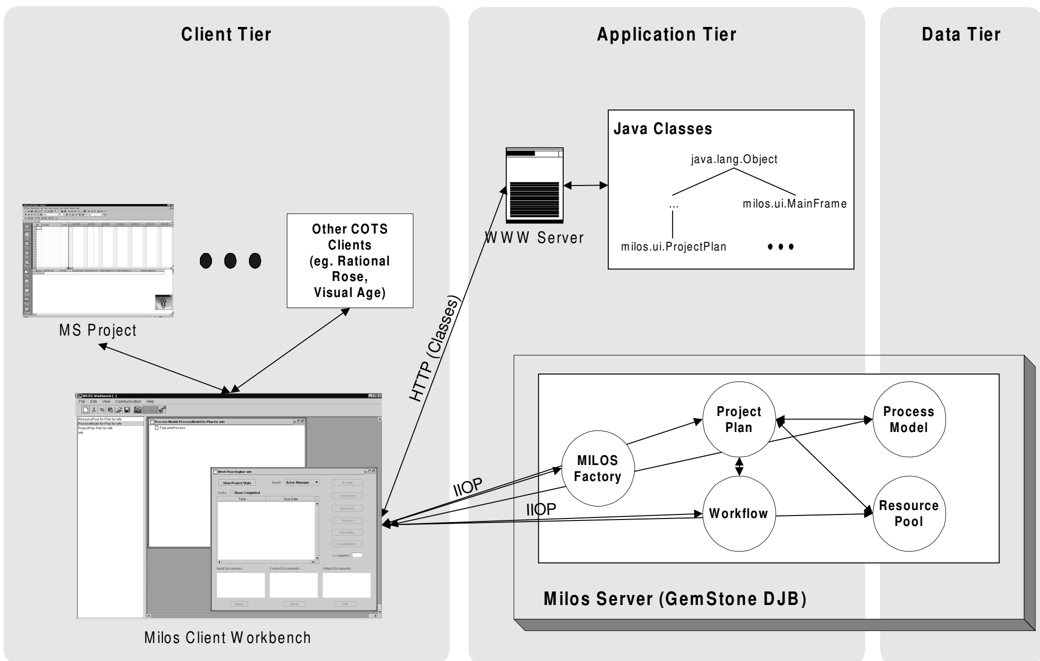


**Abbildung 1:** MILOS - GemStone Applikationsmodell

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 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

[21] 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

[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 11$^{th}$ 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.*