



1 Einleitung

Der Forschungsschwerpunkt der Arbeitsgruppe DBIS liegt bei Entwurfs- und Implementierungsfragen von Informations- und Datenbanksystemen – speziell bei Modellierungs-, Architektur- und Realisierungskonzepten zur Unterstützung von komplexen Anwendungsbereichen. Seit mehreren Jahren berücksichtigen wir verstärkt Anforderungen verschiedener Ingenieur Anwendungen und erarbeiten entsprechende DB-Konzepte; dabei konzentrieren wir uns in erster Linie auf Fragen der Datenmodellierung und der Systemarchitektur, der Unterstützung von Prozessabläufen sowie auf die Lösung von Problemen, die im Zuge von Heterogenität und der zugehörigen Forderung nach Interoperabilität auftreten. Als übergreifendes Forschungsthema werden zur Zeit objekt-relationale Datenbankkonzepte entwickelt bzw. ausgewertet. Weiterhin arbeiten wir an Fragestellungen im Zusammenhang mit dem WWW – wie die Anbindung von DBS und die DB-gestützte dynamische Aktualisierung von HTML- und XML-Dokumenten.

2 Objekt-relationale DB-Technologie

Ein Schwerpunkt der momentanen Forschungsarbeiten betrifft die Nutzung von objekt-relationaler DB-Technologie bei der Unterstützung des Softwareentwicklungs-(SE)Prozesses. Aus der Vielfalt der Entwicklungstätigkeiten ergeben sich weitreichende Anforderungen an eine gemeinsame Datenbank, einem so genannten Repository. So müssen sowohl große Mengen einfach strukturierter Daten, wie z. B. in Experimenten gewonnene Messdaten, als auch komplex strukturierte Entwurfsobjekte, wie z. B. Softwaremodule, und komplexe Dokumente, die auch Bilder und Texte umfassen können, integriert verwaltet und verarbeitet werden. Um die heterogenen Anforderungen be-

züglich Daten- und Verarbeitungsmodell angemessen erfüllen zu können, muss das verwendete DBS an das jeweilige Anwendungsprofil angepasst werden können. Hierbei verspricht vor allem das Konzept der Erweiterbarkeit um benutzerdefinierte Datentypen, Funktionen und Zugriffspfadstrukturen eine flexible Anpassung an die Anwendungsdomäne. In verschiedenen Projekten untersuchen wir, inwieweit sich relationale und objektorientierte Konzepte mit dem Ziel einer geeigneten Modellierungsmächtigkeit, ausreichender Erweiterbarkeit, angepasster Schnittstellen und angemessenem Leistungsverhalten integrieren lassen.

2.1 Generierung angepasster Repositories

Um die Kooperation und die Wiederverwendung in großen Entwicklungsumgebungen zu unterstützen, sollte ein gemeinsam genutztes Repository verwendet werden, das neben der herkömmlichen DB-Funktionalität zusätzliche Dienste, wie z. B. Checkout/Checkin, Versionierung oder Konfigurierung, bereitstellt. Mit Hilfe von SERUM (Generating Software Engineering Repository using UML) sollen für den SE-Prozess spezifische Erweiterungen konzipiert und realisiert werden, die z. B. die Versionierung von Entwurfsdaten erlauben bzw. die Zusammenarbeit der Entwurfsdatenverwaltungskomponente mit der SE-Umgebung ermöglichen.

Der SERUM-Ansatz zeichnet sich dadurch aus, dass nicht ein universeller Repository-Manager zur Verfügung gestellt, sondern vielmehr ein *Framework* entwickelt werden soll, mit dem verschiedene Repository-Manager generiert werden können, die an die Bedürfnisse spezieller Anwendungsbereiche angepasst sind. Um Dienste sowohl für die Erstellung von Repositories als auch in den von SERUM erstellten Repository-Managern in das DBVS zu integrieren, ver-

wenden wir ein objekt-relationales DBVS (ORDBVS).

Beispielsweise wird die Versionierung von Produktdaten in vielen Anwendungsbereichen benötigt. Obwohl eine Vielzahl von Versionierungssemantiken zur Unterstützung verschiedener Anwendungen mit unterschiedlichen Bedürfnissen verlangt wird, gibt es einige grundlegende Eigenschaften der Versionierung. Durch SERUM soll ein Framework mit diesen grundlegenden Versionierungseigenschaften zur Verfügung gestellt werden, das an die anwendungsspezifische Versionierungssemantik angepasst und anschließend in einer Entwicklungsumgebung eingesetzt werden kann.

2.2 Verarbeitungsmodell für ORDBVS

Eine angemessene Client/Server-Architektur sollte die reine *Server-Zentrierung* gegenwärtiger DBS aufheben, sodass ein flexibleres Verarbeitungskonzept für Entwurfsanwendungen, wie die Softwareentwicklung, angeboten werden kann. Dies umfasst neben der Bereitstellung von DBS-Komponenten zur Verwaltung eines Client-Caches auch die Erarbeitung von Konzepten und Mechanismen zur dynamischen Bestimmung des Ausführungsorts (Server oder Client) von Anwendungsfunktionen.

Die angereicherten Modellierungskonzepte zusammen mit den Möglichkeiten der Systemintegration von Funktionen/Verhalten lassen ORDBVS für komplexe Anwendungen geeignet erscheinen, die jedoch bis zu einem gewissen Grad auch eine *Client-seitige Verarbeitung* erwarten. Aus diesem Grund denken wir über eine Flexibilisierung des bisher starren Verarbeitungskonzeptes von ORDBVS nach. Wir streben in diesem Zusammenhang ein Verarbeitungskonzept an, das in der Lage ist, dynamisch und implizit über den Verarbeitungsort (Client oder Server) einer zur Ausfüh-

rung anstehenden Funktion zu entscheiden. Unser erstes Konzept beruht auf statistischen Methoden und sieht vor, den Aufwand (Verarbeitungszeit) von Funktionen zur Laufzeit zu protokollieren. Als weitere Informationen werden der erwartete, von der Funktion benötigte Verarbeitungskontext und die anfallenden Kosten zum Transport dieser Daten zum jeweiligen Verarbeitungsort betrachtet. Darüber hinaus werden Abhängigkeiten zwischen Funktionen, die eine Abschätzung des weiteren Verlaufes der Verarbeitung zulassen, als Entscheidungsgrundlage herangezogen. All diese Informationen werden zur Laufzeit ausgewertet, um dynamisch über den Verarbeitungsort einer zur Ausführung anstehenden Funktion zu entscheiden.

2.3 Erfahrungsdatenbank

Mit einer Erfahrungsdatenbank (EDB) soll Wissen und Erfahrung aus SE-Projekten in personenunabhängiger Weise gesammelt und verwaltet werden, um es den Entwicklern in nachfolgenden Projekten verfügbar machen zu können. Im Rahmen des Sonderforschungsbereichs (SFB) 501 soll eine EDB die Abwicklung von SE-Projekten nach dem Quality-Improvement-Paradigma unterstützen, das eine ablaufbezogene Vorgehensweise zur umfassenden Wiederverwendung von Software-Artefakten anbietet. *Erfahrungen* sind also im Wesentlichen in SE-Prozessen entstandene, potenziell wiederverwendbare Artefakte. Das Spektrum reicht hier von Beschreibungen/Bewertungen von in früheren Projekten genutzten Methoden und Technologien über Prozessmodelle bis hin zu konkreten Produktdaten, die in dem neuen Prozess wiederverwendet bzw. zwecks neuartiger Verwendung angepasst werden sollen.

In einer ersten Phase wurden Verfahren entwickelt, mit denen Projekterfahrungen durch Beschreibungen und Bewertungen in die EDB aufgenommen werden konnten. Dazu wurden die Erfahrungen durch so genannte Charakterisierungsvektoren annotiert und als Daten repräsentiert. Zum Wiederauffinden wurden Möglichkeiten einer ähnlichkeitsbasierten Suche bereitgestellt.

In weiteren Phasen soll eine schrittweise Integration von *Erfahrungs-, Produkt- und Prozessdaten* erfolgen. Dazu sind die in einem Projekt anfallenden Produktdaten (unter Berücksichtigung des spezifischen Produktdatenmodells) in die EDB aufzunehmen. Auf diese Weise können sie mit Erfahrungsdaten, die für das betreffende Projekt vorliegen, in Beziehung gesetzt werden, was sowohl ihre Auswertung hinsichtlich ihrer Wiederverwendung als auch ihre spätere Projektnutzung erheblich vereinfacht. Zur Bearbeitung der Produktdaten ist dann eine zugeschnittene EDB-Schnittstelle zur Verfügung zu stellen, über die Entwurfswerkzeuge auf die Produktdaten zugreifen können, die aber auch Ad-hoc-Zugriffe durch den Benutzer unterstützt. Schließlich ist es möglich, dass die EDB auch SE-Prozessmodelle (und ihre anfallenden Daten) in integrierter Weise kontrolliert und verwaltet, um sie so für die Wiederverwendung verfügbar zu halten.

2.4 VirtualMedia

Im VirtualMedia-Projekt 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, Dokumente anderer Anwendungen zu öffnen und zu bearbeiten, ohne deren Speicherungsformat zu kennen. Hiervon können vor allem Datenbanken profitieren, bei denen die Dokumente oftmals wesentlich lang-

lebiger 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 wurde auch die Anwendungsschnittstelle (in Form einer deskriptiven Anfragesprache) entworfen.

Die durch VirtualMedia angegangenen Probleme treten auch im Kontext der EDB auf. Hier ist vorgesehen, Dokumente, die von externen SE-Werkzeugen erzeugt und bearbeitet werden, in ihrem proprietären Format in der Datenbank zu verwalten. Der Nutzen der EDB steigt dann erheblich, wenn beim Browsen (beispielsweise mit einem Web-Browser) auch diese Dokumente bestmöglich dargestellt werden können, ohne dass erst das entsprechende Werkzeug oder ein spezieller Viewer auf dem Client installiert werden muss. Die durch VirtualMedia realisierte *Formatunabhängigkeit* löst dieses Problem, indem die Dokumente je nach Bedarf der anfragenden Anwendung in der jeweils optimalen externen Repräsentation ausgeliefert werden.

2.5 Erweiterbarkeit

Da in den objekt-relationalen Datenmodellen die Semantik von Beziehungstypen nur schwach unterstützt wird, ist eines unserer Forschungsziele die systematische Verfeinerung von Beziehungstypen und ihre Ausstattung mit mehr *systemkontrollierter Semantik*. Unsere Arbeiten zu dieser Thematik fassen wir unter dem Namen ORIENT (Object-based Relationship Integration Environment) zusammen.

Im diesem Projekt wurde ein Konzept zur Erweiterung von Beziehungstypen und Verfeinerung ihrer Semantik entwickelt. Zur Integration dieser Beziehungen, ihrer Operationen sowie der Kontrolle ihrer Semantik durch ein ORDBVS wurden drei verschiedene Ansätze im

Detail untersucht. Die *flache Integration* über eine Zusatzebene ohne Änderung der ORDBVS-Schnittstelle (z.B. SQL99) ist unbefriedigend, da viele Aspekte der neuen Konzepte nicht umgesetzt werden können. Es handelt sich hierbei eher um eine rein syntaktische Umsetzung, bei der das Ziel einer semantischen Anreicherung von Beziehungen weit verfehlt wird. Die *Tiefenintegration* in ein ORDBVS dagegen garantiert die angestrebte Umsetzung der neuen Konzepte und ihre effiziente Verarbeitung. Jedoch impliziert dieser Ansatz weitreichende Veränderungen und Anpassungen in allen Schichten eines ORDBVS, was nur durch den Hersteller erfolgen könnte. Die Kompromisslösung für die Integration sieht deshalb die Nutzung von *vorgegebenen System-schnittstellen* vor, die in heutigen ORDBVS als DataBlades oder Extenders 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 jedoch ein zufriedenstellendes Leistungsverhalten zu erwarten ist.

2.6 Objekt-relationaler Benchmark

Objektorientierte Programmiersprachen (OOPL), wie z. B. C++, Java, SmallTalk usw., haben sich in der Entwicklung komplexer Softwaresysteme durchgesetzt. Mit der Integration objektorientierter Konzepte und Erweiterungsmechanismen verfolgen ORDBVS das Ziel, komplexe Softwaresysteme der neuen Generation, die überwiegend mit OOPL entwickelt werden, besser und effektiver zu unterstützen. Dieses Forschungsprojekt hat zum Ziel, die Fähigkeiten eines (beliebigen) ORDBVS zur Unterstützung von OOPL (in der Entwicklung komplexer Softwaresysteme) *zu bewerten und zu quantifizieren*. Zu diesem Zweck werden zunächst Modellierungs- und operationale Aspekte sowohl des objektorientierten als auch des objekt-relationalen Paradigmas betrachtet, um die noch immer zu be-

obachtende Lücke zwischen diesen beiden Paradigmen aufzuzeigen. Die dabei getroffenen Aussagen werden dann mit Hilfe von Leistungsmessungen belegt, die mit einem neuartigen Benchmark-Ansatz durchgeführt werden, den wir zur Analyse des Leistungsverhaltens von ORDBVS unter besonderer Berücksichtigung ihrer Fähigkeiten bei der Unterstützung von OOPL in der Entwicklung komplexer Softwaresysteme entwickelt haben. Neben dem Ziel, das Leistungsverhalten von ORDBVS entsprechend quantifizieren zu können, setzt sich dieses Projekt das Langzeitziel, die weitere Entwicklung der objekt-relationalen DB-Technologie dahingehend zu beeinflussen, dass die Anforderungen der *objektorientierten Systementwicklung* angemessener als bisher erfüllt werden können.

3 Web-basierte DB-Anwendungen

Bei der Entwicklung Web-basierter DB-Anwendungen nutzen wir vor allem die Möglichkeiten von ORDBVS zur dynamischen Erweiterbarkeit des Systems um benutzerdefinierte Datentypen und Funktionen sowie die des Internet, insbesondere Plattform- und Ortsunabhängigkeit der DB-Zugriffe, um eine hohe Systemflexibilität zu erreichen. Die Schwerpunkte der momentanen Arbeiten liegen bei der Verwaltung dynamischer Dokumente in Web-Informationssystemen (WIS) und bei der Personalisierung und Integration von Web-Dokumenten in Portalen.

3.1 iWebDB

Beim Einsatz von WIS gibt es zahlreiche Probleme, wie etwa hohen Personal- oder Ressourcenaufwand, nicht aktualisierte Informationen, veraltete Navigationshilfen und Indexdaten und zunehmende Unübersichtlichkeit. Im Projekt iWebDB (integrierte Web-Datenbank) wird untersucht, wie sich durch das Zusammenspiel von ORDBS und neuen Web-Technologien wie XML die Probleme lösen bzw. reduzieren und damit ein effektiver, interner Informationsaustausch erreichen lassen. Insbesondere werden die Möglichkeiten der *Erweiterbarkeit* genutzt,

um Web-Dokumente in der gleichen Form wie vordefinierte Datentypen zu behandeln und die dazu benötigte Funktionalität in das ORDBVS zu integrieren.

Eine Prototyp-Implementierung hat bereits gezeigt, dass sich durch iWebDB viele organisatorische Anforderungen und Aufgaben effektiv unterstützen lassen. Dazu gehören einfache Administrierbarkeit, Automatisierung der Rechte- und Gruppenverwaltung, Gewährleistung der Link-Konsistenz und Gültigkeit der Dokumente sowie Bereitstellung geeigneter Suchmechanismen. Außerdem übernimmt iWebDB die *automatische Aktualisierung und Generierung* von Web-Dokumenten, sobald sich die zugehörigen, in einem ORDBS gespeicherten Informationen ändern.

3.2 Netzbasiertes Lehren und Lernen

Zeit- und raumunabhängig zu lernen, ist ein großer Vorteil netzbasierter virtueller Lernumgebungen, die dafür vielfältige Unterstützung bieten können. So kann die Möglichkeit geschaffen werden, die Lernumgebung aktiv mitzugestalten, z. B. durch Annotationen, Veränderung in der Präsentation oder Neustrukturierung der Lerninhalte. Dazu wird die Schaffung einer persönlichen, virtuellen Lernumgebung angestrebt, die eigene Materialsammlungen, Kommentare oder Informationen beinhaltet, die auch bereits bearbeiteten Aufgaben sowie eventuell deren Lösung verwaltet und die Möglichkeit bietet, diese wiederum anderen Lernenden zugänglich zu machen. Weiterführende Aspekte einer solchen Personalisierung umfassen die Möglichkeit, adaptiv auf das Verhalten Lernender einzugehen und Inhalte dem Vorwissen und dem Fortschritt in der jeweiligen Lernphase individuell anzupassen. Auskunft über das Lernverhalten sollte jederzeit möglich sein. Verschiedene Kommunikationstechniken ergänzen die Möglichkeiten der Unterstützung.

Personalisierung und Integration von Web-Dokumenten sind die wesentlichen Forschungsaspekte, die im Projekt SharX (Accessing Heterogeneous Artefacts using XML) verfolgt werden. Dabei setzt die Individualisierung von Lernumge-

bungen eine strikte Trennung von Präsentation, Struktur und Inhalt der Dokumente voraus. Unser Lösungsansatz basiert auf einem deklarativen Web-Site Management und sieht vor allem ein einheitliches Datenmodell vor, das die hierarchische Struktur einer Web-Site zu beschreiben und Beziehungen explizit zu spezifizieren erlaubt sowie eine deklarative Anfragesprache zur Verfügung stellt. Damit lässt sich eine umfassende Dokumentenarchitektur definieren, bei der die Dokumentenstruktur und ihre möglicherweise verschiedenen Repräsentationen unabhängig voneinander verändert und angepasst werden können. Die resultierende Systemarchitektur, die den kombinierten Einsatz von Integrationstechniken (für Daten und Anwendungen), Portal- und XML-Technologien erfordert, will einen »Unified View« auf alle verwalteten Dokumente bieten, was eine Abstraktion von der XML-Dokumentenstruktur und der physischen Repräsentation der enthaltenen Daten voraussetzt.

4 Anwendungsintegration

Unsere Forschungsaspekte zum Thema Anwendungsintegration umfassen vor allem Fragen der Interoperabilität und Ablaufkontrolle in einem heterogenen und verteilten Systemumfeld, wie es beim Workflow-Management in technischen Informationssystemen oder in organisationsübergreifenden Anwendungen zu erwarten ist. Außerdem entwickeln wir Lösungsansätze, wie mit deklarativen Anfragesprachen auf Daten zugegriffen werden kann, die sowohl in heterogenen Datenquellen als auch in gekapselten Anwendungssystemen gehalten werden.

4.1 Langlebige Workflows

Workflow-Management spielt bei technischen Anwendungen eine zentrale Rolle. Im Laufe eines Produktentwicklungszyklus sind viele unterschiedliche Aufgaben zu erledigen, die für einen reibungslosen Ablauf koordiniert werden müssen. Die zeitlich sehr lang laufenden *technischen Workflows* (oft mehrere Jahre) unterscheiden sich grundlegend von *klassischen Workflows*, deren aktive Lebensdauer häufig in Stunden oder Tagen

bemessen sind. Bei klassischen Workflows, die sehr oft ablaufen, wird ein festes Workflow-Schema vordefiniert, das für jeden Workflow instanziiert wird. Die so erzeugten Workflows bewegen sich exakt im Rahmen dieser Vorgaben, Abweichungen sind nicht vorgesehen und benötigen eine besondere Behandlung. Bei technischen Workflows hingegen kann eine solch detaillierte Vorplanung nicht stattfinden. Zum einen lässt sich der Ablauf auf die lange Entwicklungsperiode hin nicht exakt vorausplanen, woraus sich die Forderung nach einer ständigen Verfeinerung des Ablaufs ableitet. Zum anderen ergeben sich immer wieder Änderungen, weil sich beispielsweise die für das Endprodukt relevanten Vorschriften und Gesetze ändern. Außerdem spielen Aspekte der Heterogenität der System- und Ablaufumgebung eine vorrangige Rolle, bei der vorhandene Technologien zu integrieren und erhebliche Veränderungen während einer Workflow-Ausführung zu bewältigen sind. Technische Workflows sind deshalb eigenständige Workflows, die nur ihrem eigenen Schema genügen und in verschiedener Hinsicht anpassbar sein müssen.

Deshalb zielt das Projekt Arktis (a reliable kernel for workflows in technical information systems) darauf ab, die Modellierung, Ausführung und Kontrolle von technischen Workflows als eine Einheit zu sehen, und nicht – wie im klassischen Workflow-Management üblich – eine Trennung von Modellierung und Ausführung zu forcieren.

Das Arktis-Workflow-Modell führt das Konzept der *Service-Workflows* ein, die durch eigene Arktis-Services generiert werden. Dadurch kann das Arktis-WfMS durch neue Services erweitert werden, ohne das System selbst zu modifizieren. Arktis-Services ermöglichen auf diese Weise, auf spezielle Anforderungen zugeschnittene Abläufe zu formulieren, die der Service dann algorithmisch umsetzt. Beispiele spezieller Services in technischen Informationssystemen sind Scheduling-, Milestone-, Circular- oder MultiInstance-Service.

Die Modellierung von Abläufen sollte in natürlicher und intuitiver Weise durch den Ingenieur vor Ort erfolgen können. Diese Forderung wurde mit Hil-

fe von Makros umgesetzt, die Details von wiederkehrenden Modellierungskonstrukten vor dem Ingenieur verbergen. Durch die Nutzung solcher Makros, die in einer erweiterbaren Bibliothek gewartet werden, gestaltet sich der Entwurf eines Ablaufs recht einfach.

Die Implementierung des Arktis-WfMS erfolgt durch eine Workflow-Engine auf Petrinetz-Basis, deren Komponenten in ein ORDBVS integriert sind. Durch ihre interpretierende Arbeitsweise ist es relativ einfach, Anpassungen, Umpfanungen oder Erweiterungen an einem laufenden Workflow vorzunehmen. Weiterhin nutzt Arktis extensiv Warteschlangen für die Kommunikation zwischen den Komponenten sowie zwischen dem WfMS und den Benutzern, um so eine erhöhte Ausfallsicherheit und eine Art von Lastbalancierung zu erzielen.

4.2 Organisationsübergreifende Workflows

Um Geschäftsprozesse, die Organisationsgrenzen überspannen und in beteiligten Bereichen kohärent abgewickelt werden sollen, modellieren und rechnergestützt ausführen zu können, müssen Fragen der Heterogenität und Interoperabilität in allen ihren Modellierungs- und Ablaufaspekten explizit und übergreifend berücksichtigt werden. Da wir uns vorwiegend um die rechentechnische Umsetzung solcher »eigenständige Bereiche umspannenden« Geschäftsprozesse konzentrieren, sprechen wir von *organisationsübergreifenden oder globalen Workflows*, die durch Integration von heterogenen, lokalen (Sub-) Workflows zu bilden sind, die auf verschiedenartigen WfMS ablaufen.

Um die angestrebte Integration zu einem globalen Workflow zu erreichen, verfolgen wir mehrere Teilziele. Zunächst werden Spezifikationsmittel zur Beschreibung von globalen Workflows ausgehend von lokalen Workflows (einer Reihe vorgegebener Systeme) erarbeitet. Danach sind Koordinationsmechanismen zu entwickeln, mit deren Hilfe der Koordinator den globalen Workflow unter Zuhilfenahme der lokalen Systeme steuern kann. Neben dieser globalen Ablaufsteuerung interessiert uns auch die

Datenversorgung lokaler Systeme. So kann beispielsweise in einem lokalen Workflow zur Ausführung einer Aktivität der Zugriff auf Daten notwendig sein, die in einem entfernten Produktdatenverwaltungssystem gespeichert sind. Je nach Speicherungsform (PDVS, DBVS, Datei usw.) und Vertrauensstufe zwischen den Systemen muss das globale WfMS hierzu verschiedene Grade der Unterstützung bieten. Weiterhin soll die Möglichkeit geschaffen werden, von einer »höheren Warte« einen Überblick über den aktuellen Zustand des Gesamtprozesses zu bekommen. Diese Aufgabe fällt der Monitoring-Komponente zu. Interessante Punkte hierbei sind beispielsweise, welche Vorkehrungen auf den lokalen Systemen getroffen werden müssen, um globales Monitoring zu unterstützen. Zudem kommen natürlich auch Sicherheitsaspekte ins Spiel, da natürlich nicht jeder Benutzer alle Abläufe in allen Details sehen darf.

4.3 Daten- und Funktionsintegration

Datenintegration bezieht sich auf heterogene Datenquellen, die neben den konventionellen DBS und Dateisystemen auch ORDBS, semi-strukturierte Datenquellen (HTML, XML) usw. umfassen. Es besteht die Anforderung, diese Datenquellen möglichst flexibel, effizient und komfortabel einbinden zu können. Bisher stand die Unterstützung von Interoperabilität zwischen heterogenen Datenbanksystemen im Mittelpunkt des Forschungsinteresses. Dabei wurden Konzepte und Prototypen so genannter Föderierter DBS (FDBS) und Multi-DBS entworfen, um Datenbanken mit unterschiedlichen Datenmodellen und Schemastrukturierungen integrieren zu können.

Das Bild der heterogenen Datenbankslandschaft beginnt sich jedoch zu ändern. Oft wird eine Datenbank innerhalb eines Standardsoftware-Pakets mitgeliefert. Dabei werden das DBS und die zugehörigen Anwendungsprogramme zusammengefasst und nach außen hin lediglich eine Programmierschnittstelle zur Verfügung gestellt, die dem Benutzer den Datenzugang über *vordefinierte Funktionen* er-

lauben. Beim homogenisierten Zugriff auf heterogene Datenquellen ist deshalb neben der Daten- auch eine Funktionsintegration vorzusehen.

Da bei der Anwendungsintegration neben der Daten- auch Funktionsintegration (und vielfältige Mischformen) eingesetzt werden soll, muss ein Lösungsansatz gefunden werden, der die beiden Integrationsformen nicht getrennt betrachtet, sondern eine Kombination der beiden Konzepte ermöglicht. Da für die Datenintegration bereits einige Lösungsansätze, vor allem in Form von FDBS, vorliegen, konzentrieren wir uns auf die Integration von Funktionen.

Um beim Zugriff auf heterogene Datenquellen die Flexibilität und die Vorteile einer generischen DB-Anfragesprache (SQL) sowie einer mengenorientierten Anfrageauswertung zu erhalten, gehen wir von einem FDBS als Integrationsvehikel aus. Heterogene Datenquellen lassen sich so als Tabellen über DB-Gateways oder Wrapper der Anfrageverarbeitung zuführen. Funktionsintegration geschieht über eine FDBS-interne Schnittstelle, an der die Ergebnisse von Funktionsauswertungen als abstrakte Tabellen abgeliefert werden. Die Funktionsauswertung, die sich möglicherweise über mehrere Anwendungssysteme erstreckt, kann vom FDBS initiiert und kontrolliert oder über einen vordefinierten Workflow (eines bereitgestellten WfMS) abgewickelt werden. Als konkreten Mechanismus zur FDBS-Anbindung der Anwendungssysteme untersuchen wir Wrapper (SQL/MED) und benutzerdefinierte Tabellenfunktionen.

5 Rückblick und Ausblick

Die Forschungsarbeiten der AG DBIS (früher AG Datenverwaltungssysteme) zeichnen sich dadurch aus, dass die entwickelten Konzepte durch prototypische Systemimplementierungen umgesetzt und in ihrem Leistungsverhalten bewertet werden. Waren in früheren Jahren die meisten Forschungsthemen in den Bereichen DBS-Kern, Transaktionssysteme oder Mehrrechner-DBS angesiedelt, so wurde in den letzten Jahren eine »Trendwende« überdeutlich. Vorrangig werden jetzt Themen aus den Bereichen DB-

Middleware, Anwendungsintegration und DB-basierte Anwendungsentwicklung angegangen.

Rückblickend auf nahezu 25 Jahre ihres Bestehens kann festgestellt werden, dass in der AG die Wahl der Forschungsthemen, und damit die Arbeiten selbst, stark geprägt waren von der Zusammenarbeit mit der Industrie (Siemens, IBM, DaimlerChrysler, SAP u.a.) und natürlich von der Beteiligung in SFBs und DFG-Schwerpunktprogrammen (SPP). Die Gelegenheit, seit 1977 bei Industriekooperationen, seit 1983 in SFBs – zunächst im SFB 124 (VLSI-Entwurf und Parallelität) und später im SFB 501 (Entwicklung großer Systeme mit generischen Methoden) – und im SPP »Objektbanken für Experten« mitzuarbeiten, war stets ein besonderer Ansporn für die Mitarbeiter der AG. Wir hoffen, diese »Tradition« noch viele weitere Jahre fortsetzen zu können.

6 Literatur

Das Publikationsverzeichnis der AG DBIS findet sich unter www.dbis.informatik.uni-kl.de/pubs/ Dort sind alle wesentlichen Publikationen - Bücher, Zeitschriften und Tagungsbeiträge, Habilitations- und Promotionschriften, Diplom- sowie Studien- und Projektarbeiten - von 1975 bis heute nach Erscheinungsjahr aufgelistet. Ein Großteil der Zeitschriften- und Tagungsbeiträge etwa der letzten zehn Jahre ist online zum Herunterladen verfügbar.

Theo Härder
Universität Kaiserslautern
Fachbereich Informatik
Postfach 3049
67653 Kaiserslautern
haerder@informatik.uni-kl.de

Datenbank-Spektrum ist das offizielle Organ der Fachgruppe Datenbanken (FG 2.5.1) der Gesellschaft für Informatik e.V.. Die Zeitschrift versteht sich als ein Medium für den Austausch von Informationen aus der Datenbankpraxis und -forschung. Dies soll durch die Veröffentlichung von Beiträgen erreicht werden, die sich mit Forschungsergebnisse und Praxiserfahrungen sowie mit aktuellen Standards und Technologien aus dem Bereichen Datenbanken und Datenbankanwendungen beschäftigen.

Mit einem Spektrum an Beiträgen von Einführungen und Tutorials bis hin zu wissenschaftlichen Fachbeiträgen sollen sowohl Leserinnen, die an aktuellen Forschungsergebnissen interessiert sind, als auch eher praxisorientierte Leserinnen aus der Industrie angesprochen werden, die sich neue Entwicklungen im Datenbankumfeld informieren möchten. Auf diese Weise soll der Austausch von Anforderungen und Ideen zwischen Wissenschaft und Praxis gefördert werden.

Als Fachgruppen-Organ informiert das Datenbank-Spektrum auch über Konferenzen, Messen und Workshops, Produktankündigungen, neue Produkte oder Bücher. Geplante Themen sind dabei:

- Wissenschaftliche Beiträge
- Industriebeiträge
- Standards und Tutorials
- Produktanwendungen und Lösungen
- Buchbesprechungen und Tagungsberichte
- Messe- und Tagungskalender

2 Layout-Details

Beiträge für die Zeitschrift »Datenbank-Spektrum« sollten nach Vorgaben des `dpunk.verlag` gesetzt werden. Hierzu

steht eine Schablone für FrameMaker zur Verfügung.

Das Format der Beiträge ist dreispaltig. Als Schriftart für den Haupttext wird Times in der Größe 9.5pt verwendet.

Die einzelnen Abschnitte werden auf der ersten und zweiten Gliederungsebene numeriert. Der erste Absatz nach einer Überschrift ist nicht eingerückt, alle weiteren Absätze sind jeweils mit Einzug zu versehen.

Als Anführungszeichen sind die Zeichen »Anführung« (Eingabe unter Windows mit ALT+0187 bzw. ALT+0171) zu verwenden, Hervorhebungen werden in dieser *Form* dargestellt.

3 Abbildungen

Abbildungen können über mehrere Spalten ausgedehnt werden, vorzugsweise am Seitenanfang oder -ende. Die Bildbeschriftung ist wahlweise seitlich (bei Abbildungen, die sich über mehr als zwei Spalten erstrecken) bzw. unter der Abbildung (bei kleineren Abbildungen) anzuordnen (siehe auch nächste Seite).

4 Andere Formate

Autoren, die kein Zugriff auf FrameMaker zur Erstellung ihrer Beiträge haben, können auch ein Dokument in ASCII- oder MS-Word-Format einreichen. Hierbei sollte aber zusätzlich eine Postscript- bzw. PDF-Version des Beitrages mitgeliefert werden, um einen Vergleich speziell für Formeln, Tabellen usw. zu ermöglichen.

Abbildungen sind in diesem Fall als EPS-Dateien einzureichen.

5 Literatur

[Heuer, Saake 1999] Heuer, A.; Saake, G.: Datenbanken: Implementierungstechniken. MITP-Verlag, Bonn, 2001.

Kai-Uwe Sattler
Universität Magdeburg
Fakultät für Informatik
Postfach 4120
39106 Magdeburg
kus@iti.cs.uni-magdeburg.de