

Lehrstuhl für Dialogorientierte Systeme¹ Universität Passau

1 Zielsetzung und Überblick

In der Forschung verfolgen wir das allgemeine Ziel, Datenbanktechnologie für fortgeschrittene Anwendungen insbesondere aus dem betriebswirtschaftlichen Bereich hinsichtlich Performanz, Skalierbarkeit, Fehlertoleranz und Flexibilität weiterzuentwickeln. Die aktuellen Projekte des Lehrstuhls werden zunächst kurz skizziert und danach etwas detaillierter beschrieben.

In den Projekten *ObjectGlobe* und *QueryFlow* befassen wir uns mit der Realisierung flexibler (offener) und weitläufig verteilter Anfrageauswertungsdienste im Internet. Das Ziel dieser Projekte ist es, im Internet genauso einfach Funktionalität beispielsweise in Form von Anfrageoperatoren zur Verfügung stellen zu können, wie man heute bereits Daten publizieren kann. Klienten einer Föderation von *ObjectGlobe*-Servern sind damit in der Lage, komplexe Anfragen zu initiieren, die verteilt im Internet ablaufen und dabei auf Operatoren und Daten aus verschiedenen Quellen zurückgreifen.

ServiceGlobe stellt eine Weiterentwicklung des *ObjectGlobe*-Ansatzes dar. Während wir uns in *ObjectGlobe* auf die Ausführung verteilter Anfragen konzentrierten, wird in *ServiceGlobe* die verteilte, fehlertolerante, kontext-abhängige Ausführung von Web-Services untersucht. Genau wie *ObjectGlobe* basiert *ServiceGlobe* auf einer »flächendeckenden« Verteilung von Servern, die mobilen Web-Service-Code ausführen können. Im Rahmen des *TeraByte*-Projekts kooperieren wir seit 1999 mit der SAP AG (Kooperationspartner Wolfgang Becker,

Dr. Ulrich Marquard, Dr. Axel Herbst). Das Ziel des Projekts besteht in der Erforschung von Optimierungsansätzen für SAP-Anwendungen mit einem Datenbankvolumen von mehreren Terabyte. Die bislang untersuchten/entwickelten Optimierungsverfahren umfassen die effiziente Archivierung kombiniert mit Bulk-Delete-Techniken, die Datenbank-Partitionierung sowie die XML-Datenarchivierung mit Browser und Web-Service-Schnittstellen.

Projektübergreifend haben wir in den letzten Jahren eine Vielzahl von *Anfrageoptimierungstechniken* entwickelt. Unter anderem wurden der Blackboard-Optimierer, die heuristische Join-Optimierung, die Optimierung von disjunktiven und allquantifizierten Anfragen, die Semi-Join-basierte Optimierung von verteilten Anfragen, die Optimierung von funktionalen Joins, die Optimierung von ordnungserhaltenden Hash-Joins, die frühzeitige Datenpartitionierung durch generalisierte Hashteam und die Optimierung von Bestmatch-Joins auf Datenströmen erforscht. Die aktuellen Arbeiten gehen in die Richtung der Optimierung von Anfragen über (XML-)Datenströmen.

2 Aktuelle Forschungsprojekte

2.1 ObjectGlobe

Mit dem *ObjectGlobe*-System verfolgen wir das Ziel einer verteilten *Informationsökonomie*. Die Grundlage dafür ist es, leistungsfähige Anfragebearbeitung (inklusive der Funktionalität, die man aus traditionellen Datenbanksystemen kennt) im Internet verfügbar zu machen. Die zu Grunde liegende Idee besteht darin, einen offenen Marktplatz für verschiedene Arten von Dienstleistungen zu schaffen: *Data-Provider* bieten Daten an, *Function-Provider* stellen (Anfrage-)Operatoren zur Datenbearbeitung zur Verfügung und *Cycle-Provider* führen Anfrageoperatoren aus. Diese Trennung ist dabei

nicht zwingend, d.h. ein Rechner kann gleichzeitig als *Data*-, *Function*- und *Cycle-Provider* agieren. *ObjectGlobe* ermöglicht es, komplexe Anfragepläne verteilt, und damit parallel, auszuführen. Solche Anfragepläne können die Abarbeitung von Operatoren verschiedener *Function-Provider* auf mehreren *Cycle-Providern* gleichzeitig unter Verwendung von Daten mehrerer Datenquellen erfordern. Durch den dynamischen Einsatz von externen Operatoren stellt das System allerdings auch hohe Ansprüche hinsichtlich Qualität und Sicherheit der externen Operatoren.

Wir haben das *ObjectGlobe*-System aus zwei Gründen komplett in Java implementiert: Zum einen ist Java plattformunabhängig, so dass *ObjectGlobe* auf vielen verschiedenen Systemen ohne großen Aufwand installiert werden kann. Der andere Grund ist, dass Java ein Sicherheitssystem anbietet, mit dessen Hilfe es möglich ist, die Kernfunktionalität von *ObjectGlobe* auf sichere Art und Weise um externe Operatoren zu erweitern. Diese Operatoren können sehr einfach in Java implementiert werden und werden in ihrem eigenen »Sandkasten« ausgeführt.

Quality-of-Service. Bei der verteilten Anfrageauswertung - insbesondere auf Internet-Datenquellen - stellt sich unweigerlich die Frage nach der zu erzielenden *Dienstgüte (Quality-of-Service)*. Die Qualität der Anfragebearbeitung umfasst mehrere Dimensionen, u.a. die Kosten der Anfragebearbeitung, die Antwortzeit und die Ergebnisqualität. Traditionelle Datenbanksysteme verfahren nach dem *Best-Effort-Prinzip*. Diese Systeme erhalten eine Anfrage, optimieren diese nach einem bestimmten Kriterium (z.B. Antwortzeit oder Kosten) und führen dann die Anfrage aus. Trotz Optimierung gibt es hierbei keinerlei Zusicherungen über Antwortzeiten, Kosten der Anfrageauswertung und Qualität der Ergebnisse.

Eine Auswertung von Anfragen nach dem *Best-Effort-Prinzip* ist für viele Anwendungen nicht ausreichend, z.B. Anwendungen im Bereich des e-Business. In diesem Bereich könnten Benutzer z.B. bereit sein, für Information zu bezahlen, doch nur wenn die Information innerhalb einer bestimmten Zeit zur Verfügung gestellt wird und wenn bestimmte zusätzli-

1. Zur Forschungsgruppe gehören bzw. gehörten die folgenden »hauptamtlichen« Mitarbeiter: Dr. R. Braumandl (Allianz), Dr. J. Claussen (SAP), Dr. A. Eickler (Nokia), Dr. C. Gerlhof (IBM), M. Keidl, Prof. Dr. D. Kossmann (TU München), Dr. N. Krivokapic (IBM), Dr. K. Peithner (West LB), S. Seltzsaam, B. Stegmaier, Dr. M. Steinbrunn (Gartner), Dr. K. Stocker (BMW), C. Wiesner, B. Zeller. Die Projektzuordnung ergibt sich aus den Autorenlisten der Veröffentlichungen, die man online unter der URL <http://www.db.fmi.uni-passau.de/publications> findet.

che Eigenschaften erfüllt sind, beispielsweise ausreichende Aktualität und Vollständigkeit der Ergebnisse.

Damit ein Informationssystem wie ObjectGlobe den Ansprüchen solcher Anwendungen genügen kann, sind eine Reihe von Erweiterungen erforderlich. Zunächst muss die Benutzungsschnittstelle (oder Anfragesprache) so erweitert werden, dass Benutzer und Anwendungen zusätzlich Qualitätsparameter zu ihren Anfragen spezifizieren können. Beispielsweise sollte ein Benutzer, der nach Flugverbindungen von München nach Seattle sucht, verlangen können, dass mindestens 50 Prozent aller Fluglinien bei der Suche berücksichtigt werden. Außerdem muss das System die Bearbeitung der Anfrage, d.h. alle relevanten Ressourcen (Netzwerk, Rechner, Software), permanent überwachen. Ziel dabei ist es, so früh wie möglich einzuschreiten und nach Alternativen zu suchen, falls eine Verletzung der Anforderungen des Benutzers droht, und gegebenenfalls sogar die Bearbeitung abzubrechen. In ObjectGlobe wurde ein solches *Quality-of-Service-Management* in die Anfragebearbeitungskomponente integriert, das über das Regelwerk eines Fuzzy Controllers adaptiert werden kann¹.

Anfrageoperatoren für Datenströme. Die Daten von Internet-Datenquellen werden in der Regel als (evtl. unendlich lange) Datenströme geliefert, d.h. die Daten können nur Tupel für Tupel gelesen werden. Zur Berechnung von Anfragen auf Datenströmen sind daher neue Techniken erforderlich, die z.B. eine nicht blockierende Ausführung ermöglichen und beschränkten Speicherbedarf (bei unendlich langen Datenströmen) aufweisen. In diesem Zusammenhang arbeiten wir derzeit an folgenden Themen: Semantik von Datenströmen und Anfragen auf Datenströmen, neue Operatoren und Funktionalitäten im Bereich XML-Datenströme, Semantik und effiziente Implementierung von Operatoren aus dem relationalen Umfeld im Kontext von XML, approximative Anfrageauswertung und Qualität von Anfrageergebnis-

sen.

Als ersten Operator für Datenströme wurde der Bestmatch-Join-Operator (BMJ-Operator) entwickelt, der Paare von Objekten verknüpft, für die es - hinsichtlich mehrerer Vergleichsdimensionen - kein besseres Paar gibt. Mit der Einführung von vernünftigen Einschränkungen für die BMJ-Operatoren wird die Qualität des Ergebnisses weiter verbessert und die Bearbeitungsgeschwindigkeit gesteigert. Normalerweise sind BMJ-Operatoren inhärent blockierend und deshalb nicht anwendbar auf Datenströme. Ein neu entwickelter Algorithmus basierend auf synchroner Verschiebung von Datenfenstern über die Datenströme ermöglicht eine nicht blockierende Ausführung der BMJ-Operatoren unter Verwendung der oben genannten Einschränkungen in Verbindung mit physikalischen Eigenschaften der Datenströme, z.B. Sortiertheit in einer Vergleichsdimension.

2.2 ServiceGlobe

Das ServiceGlobe-Projekt beschäftigt sich mit dem flexiblen und fehlertoleranten Einsatz von Web-Services auf Seiten des Benutzers, des Dienstentwicklers und des Dienstbetreibers. Die Basis dafür bildet die ServiceGlobe-Dienstplattform. Sie unterstützt mobilen Code, d.h. (dynamische) Dienste können zur Laufzeit auf beliebige Internet-Server, die an das ServiceGlobe-System angeschlossen sind, übertragen und dort instantiiert werden. Funktionalität, wie sie für Dienstplattformen Standard ist, z.B. SOAP-basierte Kommunikation, ein Transaktionssystem, sowie ein Sicherheitssystem, wurde bereits in ServiceGlobe integriert, bildet aber nicht den Fokus dieses Projekts.

Den Schwerpunkt des ServiceGlobe-Projekts bildet die Untersuchung von neuen Technologien für die Ausführung und den Einsatz von Web-Services in dynamischen Umgebungen, die im ServiceGlobe-System implementiert werden. Zurzeit werden dabei die vier folgenden Themen in diesem Zusammenhang bearbeitet.

Dynamische Dienstausswahl. Die dynamische Dienstausswahl ermöglicht es, Dienste zur Laufzeit basierend auf ei-

ner technischen Spezifikation des gewünschten Dienstes dynamisch auszuwählen und aufzurufen. Sie erlaubt damit beim Aufruf eines Dienstes eine Abstraktion von den tatsächlichen Web-Services. Mit Vorgaben bezüglich der Metadaten von Web-Services kann die dynamische Dienstausswahl gesteuert werden. Außerdem können Antworten auf Eigenschaften überprüft und gegebenenfalls eliminiert werden. Mit Vorgaben kann auch festgelegt werden, wie viele und auf welche Weise Web-Services aufgerufen werden sollen. Die Verwendung von dynamischer Dienstausswahl bei der Entwicklung von Web-Services erhöht die Fehlertoleranz eines Dienstes, indem die tatsächlich aufzurufenden Web-Services erst zur Laufzeit bestimmt werden. Nicht mehr existente Dienste werden dabei nicht mehr angesprochen, neue Dienste werden automatisch in die Auswahl miteinbezogen. Sollte beim Aufruf eines Dienstes ein Fehler auftreten, wird automatisch ein anderer Dienst aufgerufen, sofern einer verfügbar ist.

Kontext für Web-Services. Die Verwendung von Kontexten ermöglicht es, Benutzern eine angepasste und personalisierte Version der Web-Services zur Verfügung zu stellen. Die Integration von Vorgaben für die dynamische Dienstausswahl in die Kontexte von Web-Services stellt eine Möglichkeit zur Personalisierung von Web-Services dar, indem diese Vorgaben von der Dienstplattform automatisch berücksichtigt werden. Damit kann ein Benutzer einen Web-Service z.B. dazu veranlassen, nur kostenlose Web-Services aufzurufen. Weitere Möglichkeiten ergeben sich durch die Integration von Informationen über die Benutzer von Diensten oder Sicherheitsinformationen, die im Rahmen der Authentifizierung/Autorisierung von Web-Services Verwendung finden.

Generischer Dispatcher. Der generische Dispatcher bietet Betreibern von Diensten die Möglichkeit, einen Dienst auf mehreren Rechnern zur Verfügung zu stellen. Der Dispatcher nimmt dabei alle Anfragen entgegen und verteilt diese auf die verschiedenen Dienstinstanzen. Der Dispatcher kann bei vielen bereits existierenden Diensten eingesetzt werden, ohne dass dafür die Implementierung des

1. Für die Quality-of-Service-Arbeiten hat Dr. R. Braumandl den Dissertationspreis der BTW-Tagung 2003 erhalten.

Dienstes geändert werden muss. Die Verwendung mehrerer Dienstinstanzen erhöht die Last, die von einem Dienst bewältigt werden kann, reduziert die Antwortzeit und verbessert die Verfügbarkeit eines Dienstes, da der Ausfall eines Rechners vom Dispatcher registriert wird und weitere Anfragen nur noch an die funktionierenden Rechner weitergeleitet werden. Eine Besonderheit des Dispatchers ist die automatische Dienstreplikation. Diese erlaubt es, Dienste mit Ressourcenbedarf (z.B. Datenbank) dynamisch auf dem Dispatcher zur Verfügung stehenden Rechnern zu installieren und geeignet zu konfigurieren.

SOAP-Caching. Durch das semantische Caching von SOAP-Anfragen kann man die Leistungsfähigkeit SOAP-basierter Systeme steigern. Dazu wird die Semantik der Anfrageparameter in geeigneter Weise in XML spezifiziert. Dadurch ist es dem SOAP-Cache nicht nur möglich, Anfragen zu beantworten, die vorher exakt so schon einmal gestellt wurden, sondern er kann auch ähnliche Anfragen beantworten. Für die Cache-Konsistenz stehen verschiedene Strategien zur Verfügung, die von einfachem TTL (Time-to-Live) bis zur Invalidierung von Cache-Einträgen durch einen vom Betreiber eines Dienstes zur Verfügung gestellten Invalidierungs-Dienst reichen. Der Betreiber eines Dienstes kann steuern, ob und wie die gelieferten Daten in einem Cache gespeichert werden dürfen, indem er einen SOAP-Header in seine Antwort einfügt.

2.3 Das QueryFlow Projekt

Virtuelle elektronische Marktplätze und virtuelle Unternehmen entwickeln sich zu einer wichtigen Anwendung der Anfragebearbeitung. Es erfordert eine hoch flexible, verteilte Anfragebearbeitung, um einen skalierbaren Marktplatz mit hunderten oder tausenden teilnehmenden Anbietern zu realisieren. Die traditionelle Art, einen virtuellen Marktplatz aufzubauen, ist es, in einem zentralen Data Warehouse *alle* Daten der teilnehmenden Unternehmen zu speichern. Dieser Ansatz hat allerdings mehrere Nachteile: Die Sicherheit von vertrauenswürdigen Daten ist gefährdet, da diese Daten nicht

mehr unter der direkten Kontrolle der Anbieter sind. Die Kohärenz von dynamischen Daten, wie Preis und Lieferinformation ist verletzt, da der Marktplatz veraltete Daten enthalten kann. Außerdem ist eine aufwendige Schemaintegration notwendig und sämtliche Daten aller Anbieter müssen in ein gemeinsames Format konvertiert werden. Oft wäre es einfacher, die Daten bei den Teilnehmern zu belassen und lokal Teilanfragen auszuwerten. Dabei werden Daten nur auf Anforderung konvertiert und sind immer aktuell. Auf materialisierten Daten sind nur beschränkte Datenbankoperationen möglich; so ist z.B. eine dynamische und individuelle Preisbildung mit Eingriffsmöglichkeiten durch den Anbieter nicht möglich.

HyperQueries. Der Einsatz von *HyperQueries* bietet eine neue Möglichkeit zur dynamischen, verteilten Anfragebearbeitung im Internet. HyperQueries sind im Wesentlichen Anfragepläne, auf die durch Hyperlinks verwiesen wird. Dadurch agiert der Marktplatz als Vermittler zwischen Kunden, die eine Anfrage stellen, und Anbietern, die ihre Unteranfragen ausführen. Die Hyperlinks, welche die HyperQueries referenzieren, sind als (virtuelle) Attribute in die Datenobjekte des Marktplatzes eingebettet. Der Zugriff auf ein virtuelles Attribut stößt automatisch die Berechnung des Wertes am Rechner des Anbieters an. Damit bleiben alle vertrauenswürdigen Daten unter der Kontrolle des Anbieters. Die Ausführung der HyperQueries erfolgt dynamisch und basiert nur auf den Hyperlinks, d.h. es werden nur die Pläne ausgeführt, die benötigt werden.

Eine Referenzarchitektur für Marktplätze. Basierend auf diesen HyperQueries entwickeln wir eine Referenzarchitektur zum Aufbau von skalierbaren, verteilten B2B Marktplätzen. Die Herausforderungen eines elektronischen Marktplatzes beinhalten u.a. Transparenz und Skalierbarkeit, d.h. neue Teilnehmer können leicht dem Marktplatz beitreten, ohne die Anfragen anderer Teilnehmer zu beeinflussen. Weitere wichtige Aspekte unseres Ansatzes sind die Integration von existierenden Systemen, die Unterstützung von offenen Standards (XML, SOAP, XML Signature, etc.), Sicherheit,

Fehlertoleranz und die Administration des Marktplatzes. Die Teilnehmer sind in der Lage, ihre Daten am Marktplatz selbstständig ohne die Unterstützung durch den Marktplatzadministrator zu verwalten. Unsere Referenzarchitektur basiert auf offenen Standards und bietet Java-basierte, erweiterbare Web-Services zur Administration des Marktplatzes, z.B. zur Registrierung und Aktualisierung von Kunden, Anbietern und Produkten.

2.4 Das Terabyte-Projekt

Das Terabyte-Projekt der SAP AG dient der Untersuchung und Entwicklung neuer Technologien, um den Anforderungen von (verteilten) SAP-Anwendungen mit mehreren Terabyte an Daten gerecht zu werden. Die Kooperation der SAP AG mit unserer Forschungsgruppe verfolgt das Ziel, die notwendige Grundlagenforschung im Bereich der betriebswirtschaftlichen Datenbanksysteme bereit zu stellen. Die Leistung von Systemen mit mehreren Terabyte Daten entspricht oft nicht den Anforderungen und auch die Wartung ist oft nicht mehr in den Wartungszeitfenstern möglich. Wir haben deshalb zum einen untersucht, wie sich die Datenmengen in den Datenbanken der betriebswirtschaftlichen Anwendungen verringern lassen (*Archivierung*) und wie zum anderen die verbleibenden Daten effizient verwaltet werden können (*Partitionierung*). Um die Veränderungen in der Leistung der Datenbanken messen zu können wurde ein entsprechender Benchmark implementiert (*Leistungsanalyse*).

Archivierung. Zur Verringerung der Datenmenge wird in SAP-Systemen *Datenarchivierung* eingesetzt, d.h., Daten von selten benötigten betriebswirtschaftlichen Objekten werden aus den OLTP Datenbanksystemen auf Tertiärspeichersysteme verschoben. Dadurch wird das Datenvolumen der Datenbank reduziert und damit die Leistung des Datenbanksystems erhöht und die Wartbarkeit verbessert.

Der zeitaufwendigste Arbeitsschritt bei der Archivierung ist das Löschen der Daten aus der produktiven Datenbank. Die standardmäßig verwendete »tuple-at-a-time«-Löschmethode führt zu vielen

zufälligen Zugriffen auf den Hintergrundspeicher und belastet dadurch den Datenbankrechner stark. Wir haben deshalb eine neue Methode (*Bulk-Delete*) zum massenhaften Löschen von Einträgen aus einer relationalen Datenbank entwickelt, die bis zum Faktor 10 schneller ist, indem sie sequenziell auf den Hintergrundspeicher zugreift.

Bei der von SAP eingesetzten Art der Datenarchivierung werden die Daten auf Anwendungsseite verarbeitet und in einem proprietären, nur von einem SAP System aus bearbeitbaren Format abgelegt. Wir haben deshalb einen XML-Archivierungs-Operator entwickelt, der sehr einfach in relationale Datenbanken integriert werden kann. Die zu archivierenden Daten werden *innerhalb* der Datenbank verarbeitet und liegen nach der Verarbeitung in XML vor. Dadurch können auch andere Anwendungen auf die archivierten Daten zugreifen, z.B. zu Auswertungs-Zwecken innerhalb eines Data Warehouses. Um auch bei der XML-Archivierung auf die Vorteile der Bulk-Delete-Löschmethode zurückgreifen zu können, haben wir eine spezielle *light*-Variante dieses Algorithmus entwickelt.

Partitionierung. Wir haben in diesem Zusammenhang die *horizontale Partitionierung* von betriebswirtschaftlichen Daten untersucht. Partitionierung wurde bereits im Zusammenhang mit verteilten Datenbanken untersucht und die möglichen Leistungssteigerungen durch Partitionierung sind in diesem Zusammenhang bekannt. Allerdings sind diese Ergebnisse nicht ohne weiteres auf eine zentrale Datenbank übertragbar. Im Falle

der verteilten Datenbanken dominieren die Übertragungskosten zwischen den einzelnen Knoten die Gesamtkosten. Die Berechnungen können sehr gut parallelisiert werden, da die jeweiligen lokalen Rechenkapazitäten der Knoten genutzt werden können. Im zentralen Fall wird auf einen gemeinsamen Hauptspeicher zugegriffen. Außerdem sind Ressourcen wie CPUs oder Hintergrundspeicher sehr stark begrenzt und können sehr schnell zu Engpässen werden. Wir haben deshalb einen Benchmark entwickelt, mit dem die leistungskritischsten Operationen eines SAP R/3 Systems simuliert werden. Mit Hilfe dieses Benchmarks wurden verschiedene Partitionierungsmethoden mit einem nicht-partitionierten Tabellenlayout verglichen. Die Messungen zeigen, dass zwar in einzelnen Bereichen Leistungseinbußen hingenommen werden müssen (Updates), aber die Vorteile eines partitionierten Tabellen-Layouts diese Nachteile überwiegen.

Leistungsanalyse. Um die Auswirkungen von Tuning-Maßnahmen bewerten zu können und die Leistung bestehender SAP R/3 Systeme zu messen, wurde der TPC-D Benchmark bereits sehr früh auf das R/3 System von SAP portiert und Messungen in einem Testsystem durchgeführt. Diese Umsetzung des TPC-D Benchmarks auf SAP R/3 wurde später in SSQJ, einem SAP-internen Tool zur Qualitätskontrolle und Evaluierung neuer Technologien, integriert und steht nun sowohl Datenbankherstellern als auch den Entwicklern von SAP selbst als Meßgrundlage zur Verfügung.

Ausblick: Verteilte Anwendungen.

Bei modernen, weltweit operierenden Unternehmen können schon lange nicht mehr alle Prozesse in einer Anwendung integriert werden. Es gibt vielmehr eine Vielzahl eigenständiger Anwendungen, die miteinander kooperieren und als Ganzes die IT-Struktur des Unternehmens darstellen (siehe z.B. mysap.com Marktplätze). Unterstützt wird dieser Trend durch billige und einfach zu skalierende Rechenanlagen (Server Blades, SAN, Fibre-Channel), die es dem Unternehmen ermöglichen, sehr schnell neue Anwendungen zu integrieren und bestehende zu erweitern. Damit sich auch eine solche Ansammlung von Anwendungen nach außen hin als ein einheitliches System präsentieren kann, sind neue Ansätze im Bereich der Sicherungs- und Recovery-Techniken und des Monitorings erforderlich.

3 Veröffentlichungen

Die Veröffentlichungsliste des Lehrstuhls findet man unter der URL

<http://www.db.fmi.uni-passau.de/publications>

Die neueren Veröffentlichungen stehen dort auch für das Herunterladen zur Verfügung.

Alfons Kemper
Lehrstuhl für Dialogorientierte Systeme
Fakultät für Mathematik und Informatik
Universität Passau
Innstrasse 30
94030 Passau



Besuch des Lehrstuhls mit Studierenden beim Kooperationspartner SAP (Sommer 2002)