

LEHRSTUHL FÜR
ALLG. BWL UND WIRTSCHAFTSINFORMATIK
UNIV.-PROF. DR. HERBERT KARGL

Schwickert, Axel C.; Grimbs, Michael G.

**Realisierungsaspekte
des Client/Server-Konzepts**

ARBEITSPAPIERE WI
Nr. 3/1996

Schriftleitung:
Dr. rer. pol. Axel C. Schwickert

Information

- Reihe:** Arbeitspapiere WI
- Herausgeber:** Univ.-Prof. Dr. Axel C. Schwickert
Professur für BWL und Wirtschaftsinformatik
Justus-Liebig-Universität Gießen
Fachbereich Wirtschaftswissenschaften
Licher Straße 70
D – 35394 Gießen
Telefon (0 64 1) 99-22611
Telefax (0 64 1) 99-22619
eMail: Axel.Schwickert@wirtschaft.uni-giessen.de
<http://wi.uni-giessen.de>
- Bis Ende des Jahres 2000 lag die Herausgeberschaft bei:
- Lehrstuhl für Allg. BWL und Wirtschaftsinformatik
Johannes Gutenberg-Universität Mainz
Fachbereich Rechts- und Wirtschaftswissenschaften
Welderweg 9
D - 55099 Mainz
- Ziele:** Die Arbeitspapiere dieser Reihe sollen konsistente Überblicke zu den Grundlagen der Wirtschaftsinformatik geben und sich mit speziellen Themenbereichen tiefergehend befassen. Ziel ist die verständliche Vermittlung theoretischer Grundlagen und deren Transfer in praxisorientiertes Wissen.
- Zielgruppen:** Als Zielgruppen sehen wir Forschende, Lehrende und Lernende in der Disziplin Wirtschaftsinformatik sowie das IuK-Management und Praktiker in Unternehmen.
- Quellen:** Die Arbeitspapiere entstanden aus Forschungsarbeiten, Diplom-, Studien- und Projektarbeiten sowie Begleitmaterialien zu Lehr- und Vortragsveranstaltungen des Lehrstuhls für Allg. Betriebswirtschaftslehre und Wirtschaftsinformatik Univ. Prof. Dr. Herbert Kargl an der Johannes Gutenberg-Universität Mainz.
- Hinweise:** Wir nehmen Ihre Anregungen und Kritik zu den Arbeitspapieren aufmerksam zur Kenntnis und werden uns auf Wunsch mit Ihnen in Verbindung setzen.
Falls Sie selbst ein Arbeitspapier in der Reihe veröffentlichen möchten, nehmen Sie bitte mit dem Herausgeber (Gießen) unter obiger Adresse Kontakt auf.
Informationen über die bisher erschienenen Arbeitspapiere dieser Reihe und deren Bezug erhalten Sie auf dem Schlußblatt eines jeden Arbeitspapiers und auf der Web Site des Lehrstuhls unter der Adresse <http://wi.uni-giessen.de>

Arbeitspapiere WI Nr. 3/1996

Autoren: Schwickert, Axel C.; Grimbs, Michael G.
Titel: Realisierungsaspekte des Client/Server-Konzepts

Zitation: Schwickert, Axel C.; Grimbs, Michael G.: Realisierungsaspekte des Client/Server-Konzepts, in: Arbeitspapiere WI, Nr. 3/1996, Hrsg.: Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes Gutenberg-Universität: Mainz 1996.

Kurzfassung: Mit der prozeßorientierten Umstrukturierung eines Unternehmens geht eine signifikante Steigerung der Informations- und Kommunikationsintensität einher. Im Mittelpunkt der aktuellen Diskussion um angepaßte Informations- und Kommunikations-Systeme (IuK) steht das Client/Server-Konzept. Aufgrund einer engen Verzahnung mit der Unternehmensorganisation sind besonders bei den Eckpfeilern einer IuK-Strategie konstituierende Auswirkungen zu berücksichtigen. Mit der Umsetzung des Client/Server-Konzeptes in eine reale IuK-Architektur werden grundsätzliche Vorüberlegungen in den Bereichen der Software- und Hardware-Technik sowie der Migration zu verteilten Systemen erforderlich. Dezentralisierung, Objektorientierung und rationelles Software-Management (verteilt CASE, verteilte Standard-Software) auf der Basis von hochleistungsfähigen Workstations in einem Mehr-Ebenen-Konzept sowie die signifikante Mehrbelastung der Kommunikationswege sind die Problemfelder, die bei der aktiven Migration der Informations- und Kommunikationslandschaft zu berücksichtigen sind.

Schlüsselwörter: Client/Server, De-/Zentralisierung, Objektorientierung, Make or buy, Standard-Software, CASE, Open Software Foundation, Remote Procedure Call, Rechnerleistungsfähigkeit, Kommunikationsnetz, Stored Procedures, Performance, Migration, Rightsizing, Drei-Ebenen-Konzept, Legacy Systems

Inhaltsverzeichnis

1	Software-technische Realisierungsaspekte	3
1.1	Abkehr von zentralistisch geprägtem Denken	3
1.2	Objektorientierte Ansätze im Client/Server-Konzept.....	4
1.3	Make or buy	5
1.4	Software-Entwicklungs-Technik	5
1.5	Relevanz von Standards.....	6
2	Hardware-technische Realisierungsaspekte.....	8
2.1	Anforderungen an die Rechnerleistungsfähigkeit.....	8
2.2	Anforderungen an das Kommunikationsnetz.....	9
2.3	Performanz-Wirkungen	10
3	Migrations-Aspekte.....	10
3.1	Migration der Hardware.....	11
3.2	Migration der Software	12
	Literaturverzeichnis.....	14

1 Software-technische Realisierungsaspekte

Die veränderten Rahmenbedingungen im Client/Server-Umfeld führen im Vergleich zur Mainframe-dominierten Systementwicklung zu spezifischen Verhaltens- und Verfahrensweisen bei der Software-technischen Realisierung von Client/Server-Systemen. Fünf Aspekte sollen hier akzentuiert werden:

- Abkehr von zentralistisch geprägtem Denken,
- Objektorientierte Ansätze im Client/Server-Konzept,
- Make or buy,
- Software-Entwicklungs-Technik,
- Relevanz von Standards.

1.1 Abkehr von zentralistisch geprägtem Denken

Am Beispiel des Teilbereiches "Administrationstools für die unternehmensweite Rechnerverwaltung" soll aufgezeigt werden, daß die zentralistisch geprägte Denkweise grundsätzlich nicht auf die Ansätze zur Software-technischen Realisierung von Client/Server-Systemen übertragen werden sollten.

Renaud¹ beschreibt als ein zur Zeit noch existentes Problem von Client/Server-Systemen, daß die Mehrzahl der käuflich verfügbaren Software-Tools für die zentrale Fernadministration vieler Rechner in großen Client/Server-Systemen völlig ungeeignet und somit auf Unternehmensebene nicht einsetzbar sei. Die betreffenden Tools unterstellen, daß Rechner, die in Client/Server-Systemen unternehmensweit in LAN- und WAN-Installationen verbreitet sind, wie in geschlossenen LANs zentral verwaltet werden müssen. Es stellt sich aber die Frage, inwiefern in modernen Client/Server-Systemen eine zentrale Administration von Rechnern überhaupt technisch notwendig ist, d. h., inwiefern die Übertragung eines zentralistischen Organisationsdenkens hier möglich und sinnvoll ist. Bei der Umsetzung des Client/Server-Konzeptes muß grundsätzlich darüber nachgedacht werden, in welcher Organisationsstruktur DV-Abteilungen zukünftig eine optimale Unterstützung liefern werden. Wird zum Beispiel die DV-Abteilung dezentralisiert den verschiedenen Prozessen zugeordnet und verbleibt auf der zentralen Ebene nur noch eine Koordinationsstelle, dann **entfällt möglicherweise die Notwendigkeit für zentrale Administrationstools**. Die bestehende organisatorische Aufteilung Information Center auf der einen und konventionelle Systementwicklung und Betrieb auf der anderen Seite, kann durch Client/Server-Systeme eine Weiterentwicklung erfahren, in-

¹ Vgl. Renaud, P. E.: Introduction To Client/Server Systems - A Practical Guide for Systems Professionals, John Wiley & Sons Inc., New York u. a. 1993, S. 135 f.

dem eine **Zentralisierung nach dem Objekt** vorgenommen wird. Petzold und Schmitt² verstehen darunter eine Auslagerung der Funktionen Analyse, Implementierung und Betrieb von Anwendungssystemen in dezentrale Stellen. Sie verfolgen damit das Ziel, die Aufgaben, die dezentral effizienter gelöst werden können, nach dem **Subsidiaritätsprinzip** zu dezentralisieren. Bei dieser Aufteilung der Administration reichen dann unter Umständen die zur Zeit verfügbaren Tools für den LAN-Bereich wieder aus. Mit diesem Ansatz lassen sich wohl keineswegs die derzeit existenten Probleme mangelhafter, beziehungsweise mangelnder Tools im Client/Server-Bereich lösen. Es soll jedoch aufgezeigt werden, daß die Software-technische Realisierung von Client/Server-Systemen ein grundsätzlich verändertes Denken voraussetzt.

1.2 Objektorientierte Ansätze im Client/Server-Konzept

Bedingt durch ihre hochmodularen Strukturen und die Tatsache, daß Client- und Server-Prozesse auf verschiedenen heterogenen Plattformen implementiert werden, weisen diese Systeme eine hohe Komplexität auf. Um eine Reduktion dieser Komplexität zu erreichen, werden Prinzipien des Client/Server-Konzeptes mit denen der **Objektorientierung** verknüpft. Diese Verknüpfung soll am Beispiel der **Schaffung von Transparenz** der im Gesamtsystem Server-seitig angebotenen Dienste exemplarisch erläutert werden.

Transparenz ist eine wesentliche Voraussetzung für die Flexibilität der Client/Server-Systeme. Ein System ist für die Client-Prozesse transparent, wenn diese die Server-Dienste im Gesamtsystem aufrufen können, ohne daß ihnen der jeweilige Implementierungsort bekannt sein muß. Nur dann ist es möglich, Client- und Server-Prozesse flexibel zu kombinieren, partiell zu modifizieren oder auszutauschen. Naheliegend ist daher, **Client- und Server-Prozesse als gekapselte, unabhängige Objekte zu modellieren**, die die Objekte der realen Geschäftsprozesse abbilden. Die Daten und die Art der Methodenimplementierung sind dabei nur den Objekten selbst bekannt. Die Kommunikation findet via Nachrichtenaustausch statt, wobei das Client-Objekt eine Methode des Server-Objektes aufruft. Dieses Server-Objekt erbringt den Dienst und liefert ein Ergebnis zurück. Die beschriebene Transparenz zwischen den Objekten schafft der **Object Request Broker (ORB)**, ein von der unabhängigen Hersteller- und Anwendervereinigung **Object Management Group (OMG)** entwickeltes Konzept. Diesem ORB sind die Orte der Server-Objekte und deren angebotene Dienste bekannt. Ein Client-Objekt schickt im Rahmen einer Aufgabenbearbeitung einen Dienstauftrag an das Netzwerk ab. Der ORB ist dafür zuständig, daß dieser Auftrag erkannt und an ein Server-Objekt weitergeleitet wird. Für die Clients ist es beim Aufruf der Dienste nicht mehr notwendig, den Ort des anbietenden Servers zu kennen. Eingang hat die Idee des ORBs bereits in die Konzeption der Kooperativen Verarbeitung gefunden. Dort sorgt der Service-Broker für die notwendige Transparenz zwischen den auf Client- und Server-Prozesse verteilten Anwendungen.

2 Vgl. Petzold, H. J.; Schmitt, H.-J.: Organisatorische Gestaltungsmöglichkeiten des Einsatzes von Client/Server-Systemen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Heft Nr. 174, 1993, S. 84 ff.

1.3 Make or buy

Die Eigenentwicklung von Software hat in der Vergangenheit in vielen operativen Bereichen an Bedeutung verloren. Nicht zuletzt aus **Wirtschaftlichkeitsgründen** wird oft auf Standard-Software zurückgegriffen, die per Parametrisierung an die Struktur der Unternehmen (bestmöglich) individuell angepaßt wird. So bietet sich der Einsatz von **Standard-Anwendungs-Software** immer dann an, wenn Anwendungsbereiche durch gesetzliche Vorschriften oder betriebswirtschaftliche Kenntnisse als standardisiert gelten. Prinzipiell soll die Make-or-Buy-Entscheidung der Maxime: "Individualität im Bereich der kritischen Erfolgsfaktoren, Konformität im administrativen Bereich"³ folgen. Zu den standardisierten Bereichen zählt beispielsweise das Rechnungswesen; individuell strategisch wichtig hingegen ist der Forschungs- und Entwicklungsbereich.

Durch die strategische Neuausrichtung der Unternehmen und der darauf abgestimmten IuK-Unterstützung in Form von Client/Server-Systemen muß die Fragestellung von Make or Buy heute erweitert werden. Entscheidend ist nicht nur der Grad der Standardisierung im betriebswirtschaftlichen Funktionsbereich, sondern auch inwieweit die angebotene Standard-Anwendungs-Software den **Kriterien eines Client/Server-Konzeptes** entspricht. Verwendung finden sollte nur die Standard-Software, die eine **klare Trennung zwischen den drei Schichten Präsentation, Anwendung und Daten** aufweist und somit der Grundstruktur eines Client/Server-Konzeptes entspricht. Die mit dem Client/Server-Konzept einhergehende Individualisierungstendenz läßt eine Intensivierung der Eigenentwicklung von Software wahrscheinlich werden, wenn nicht alsbald auf ein breiteres Spektrum adäquater Standardlösungen zurückgegriffen werden kann.

1.4 Software-Entwicklungs-Technik

Mit der Realisierung des Client/Server-Konzeptes ergeben sich neue Anforderungen an die Werkzeuge für eine computerunterstützte Software-Entwicklung (CASE). Zwei Aspekte sind zu unterscheiden:

CASE-Systeme im Client/Server-Umfeld müssen die Verteilung von Präsentations-, Anwendungs- und Datenschicht bereits bei der Konzeption von Client/Server-Systemen unterstützen.⁴ Diese Anforderung war bei klassisch-monolithischen Anwendungen nicht gegeben. Somit tritt zu den seitens der herkömmlichen CASE-Systeme bereits abgedeckten Fragestellungen des "Was" (Analyse) und des "Wie" (Design) die Frage des "Wo" (Realisierungsort der Client- und Server-Prozesse).⁵ Unterstützung seitens der

3 Kargl, H., Controlling im DV-Bereich, 2. Aufl., Oldenbourg-Verlag, München, Wien 1994, S. 65.

4 Vgl. Hansen, W.-R.: Client/Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, in: Hansen, W.-R. (Hrsg.), Client/Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, Addison-Wesley Publishing Company, Bonn u. a. 1993, S. 26.

5 Vgl. Müller, B.; Starke, Th.: CASE für Client/Server ist mehr als CASE - Eine OLTP-Anwendung für das Kraftwerks-Service-Geschäft bei ABB, in: Computerwoche Extra, Nr. 4 vom 17.09.1993, S. 36.

CASE-Systeme wird dabei vor allem bei der Verteilung der Anwendungs- und Datenschicht benötigt. Hier muß nicht nur der zwischen Client- und Server-Prozessen anfallende Kommunikationsbedarf, der im Regelfall über ein Kommunikationsnetz abgewickelt wird, Beachtung finden und minimiert werden. Auch die Tatsache, daß verschiedene Hardware-Plattformen mit spezifischen Stärken und unterschiedlichen Ressourcen zum Einsatz kommen, gewinnt an Bedeutung. Laut Müller und Starke⁶ gibt es wohl theoretische Überlegungen zur automatisierten Verteilung der Programmkomponenten, diese sind aber in praxi nicht ausreichend erprobt. Somit haben zur Zeit existente Client/Server-CASE-Werkzeuge vor allem einen passiven, dokumentarischen Charakter.

CASE-Werkzeuge müssen nicht nur die Konzeption von verteilten Client/Server-Systemen unterstützen, sie selbst **sollten eine im Sinne des Client/Server-Konzeptes klare Schichtentrennung aufweisen** und auf unterschiedliche Plattformen portierbar sein.⁷ Dadurch wird die dezentrale Software-Entwicklung an grafischen Front-Ends im Zusammenspiel mit einem zentralen Repository ermöglicht. Erst dieses zentrale Repository ermöglicht eine Koordination verteilter Entwicklerkapazitäten und schafft die Voraussetzung für ein effektives Produkt- und Versionsmanagement.

Der technische Stand verfügbarer CASE-Werkzeuge hat somit auf die Umsetzung des logischen Client/Server-Konzeptes einen nicht zu unterschätzenden Einfluß. Ein schnelles Voranschreiten bei der Realisierung wird dabei vor allem von der Beseitigung hinreichend bekannter Medienbrüche zwischen Analyse und Design, der Portierbarkeit der entwickelten Anwendungs-Komponenten auf verschiedene heterogene Plattformen und der Unterstützung des Verteilungsproblems von Client- und Server-Prozessen abhängen.

1.5 Relevanz von Standards

Auch wenn Client/Server-Systeme in einer proprietären Umgebung entwickelt werden können, so wird in der Praxis meist Rückgriff auf allgemein anerkannte, herstellerübergreifende Standards genommen. Eine große Bedeutung kommt hier dem **Distributed Computing Environment (DCE) der Open Software Foundation (OSF)** zu, das als Konkretisierung der Prinzipien Offener Systeme eine Basis für den Aufbau einer standardisierten Architektur bietet. Bues bezeichnet das OSF/ DCE sogar als "die Grundlage für den Aufbau einer standardisierten verteilten Computerarchitektur"⁸. Zwei Problemfelder sollen an dieser Stelle exemplarisch aufgezeigt werden.

Zunächst ist auf die **fehlende Standardisierung der Schnittstellen zwischen Präsentations- und Anwendungsschicht** hinzuweisen. Durch eben diese fehlende Schnittstellenstandardisierung ist ein langfristiger Schutz der diesbezüglich getätigten Investitionen nicht gewährleistet. Setzt sich die vom Unternehmen favorisierte Oberfläche nicht durch, werden z. B. Modifikationen bei der Parameterübergabe an der Schnittstelle die

6 Vgl. Müller, B.; Starke, Th., a. a. O., S. 30 f.

7 Vgl. Müller, B.; Starke, Th., a. a. O., S. 28.

8 Bues, M.: Offene Systeme - Strategien, Konzepte und Techniken für das Informationsmanagement, Springer-Verlag, Berlin u. a. 1994, S. 94.

Folge sein. In solchen Fällen muß mit erheblichem Anpassungs- und Wartungsaufwand für die bereits im Unternehmen in dieser Verteilungsvariante realisierten Anwendungen gerechnet werden.

Ein zweites Problemfeld eröffnet sich beim **Remote Procedure Call (RPC)**, der zentralen Gestaltungskomponente der kooperativen Verarbeitung. Der Hauptvorteil des RPC liegt darin, daß er die Kommunikationsverbindung auch über ein physisches Netz für den Programmierer transparent macht. Erreicht wird die Transparenz durch die für die Kommunikation zuständigen Komponenten in Client- und Server-Prozessen. Diese Schnittstellen müssen aber im gesamten Client/Server-System zumindest kompatibel gestaltet sein, da sonst die Kombinationsmöglichkeiten der Kommunikation zwischen Client- und Server-Prozessen jeweils nur auf gleichartig gestaltete Teilsysteme beschränkt werden. Selbst wenn in der Phase des Designs und der Erstellung von Client/Server-Systemen noch in sich geschlossene Teilsysteme abgrenzbar sind, so sind diese Teilsysteme jedoch nicht statisch. Das Unternehmensumfeld weist immer wieder veränderte Situationen auf, die dazu zwingen, die Geschäftsprozesse im Unternehmen flexibel zu redefinieren. Ziel der Modularität von Client/Server-Architekturen ist gerade diese Flexibilität, um veränderte Anforderungen seitens der Prozesse DV-technisch situativ-optimal zu unterstützen.

Der **Vorteil eines Rückgriffs auf Standards** liegt in einer Vermeidung der beiden beschriebenen Problemfelder. Durch den Einsatz einer standardisierten Präsentationsoberfläche werden nicht nur die Investitionen in die Software-Produkte selbst geschützt, sondern auch das spezifische Know-how der Entwickler. Die möglichen Folgekosten einer Fehlentscheidung bei der Software-Auswahl in Form durch einen erhöhten Anpassungsaufwand werden vermieden. Eine Steigerung der Gestaltungsflexibilität für die Kooperative Verarbeitung bietet die Festlegung des RPCs auf eine standardisierte Form. So ist beispielsweise der RPC im OSF/DCE als Standard definiert und stellt eines der Fundamente dieser Architektur dar. Wird dieser RPC auf breiter Software-Anbieterfront unterstützt und von den Anwendern eingesetzt, kommt es nicht nur zu einem geringeren Koordinationsaufwand und zu einer möglichen Flexibilitätssteigerung innerhalb der Unternehmen. Es werden auch organisationsübergreifende, integrative Anwendungnetzwerke auf Basis kooperativer Verarbeitung möglich, die sich produktivitätssteigernd auf alle am Prozeß Beteiligten auswirken.

Die Ausrichtung an Standards bei der Erstellung von Client/Server-Systemen birgt jedoch auch einige **Nachteile** in sich, die wohl nicht überbewertet werden dürfen, hier aber Erwähnung finden sollen. Die Verwendung von Standards bei der Entwicklung von Client/Server-Systemen verursacht in vielen Fällen Einschränkungen, die gegen die Vorteile einer Herstellerunabhängigkeit abgewogen werden müssen. Allzu häufig sind **erhebliche Verzögerungen** einer breiten Verfügbarkeit vorteilhafter Verfahren und Techniken zu beobachten. Zwischen einer Innovation und deren Kennzeichnung als Standard klafft eine zeitliche Lücke, die durch die Schwerfälligkeit der Standardisierungsverfahren bedingt ist. Exemplarisch hierfür sei die Open Software Foundation (OSF) genannt, die für die Distributed Computing Environment-Architektur (DCE) verantwortlich zeichnet. Die OSF besteht zur Zeit aus über 300 Mitgliedern. Da jedes der Mitglieder ein kommerzielles Eigeninteresse hat und versuchen wird, die eigenen Pro-

dukte und Normen zum Standard erklären zu lassen, ist die Beschlußfindung für einen Standard äußerst zeitaufwendig und von Kompromissen begleitet. Diese Kompromisse führen dazu, daß Standards in der Regel nicht den "State-of-the-Art" darstellen. Sie repräsentieren entweder den kleinsten gemeinsamen Nenner verschiedener Anbieter oder aber das Abbild dominanter Marktmacht bestimmter Hersteller.

Schließlich tritt neben das Argument der Einschränkung des technisch Machbaren durch die Standards der Aspekt ihrer häufig **unpräzisen Definition**. Als Beispiel kann hier SQL angeführt werden, für welches sehr wohl eine ANSI-Norm besteht, die aber in verschiedenen Datenbank-Systemen unterschiedlich umgesetzt ist. Nicht zuletzt die herstellerepezifischen Abweichungen und Erweiterungen machen besonders heterogene Multi-Vendor-Umgebungen sehr schwer konzipierbar und aufwendig in der stabilen und verläßlichen Gestaltung.

2 Hardware-technische Realisierungsaspekte

Client/Server-Lösungen werden in der Praxis meist auf heterogenen Hardware-Plattformen implementiert. Die Verteilung der Client- und Server-Prozesse auf die Hardware-Infrastruktur sollte dabei gemäß der ablauforganisatorischen Unternehmensstruktur erfolgen. Dabei findet oftmals ein im Unternehmen bereits existentes Drei-Ebenen-Konzept, bestehend aus Mainframes, Abteilungsrechnern und PCs, weiterhin Verwendung. Die Auswirkungen einer solchen Verteilung von Client- und Server-Prozessen lassen sich aus folgenden Perspektiven betrachten:

- Anforderungen an die Rechnerleistungsfähigkeit,
- Anforderungen an das Kommunikationsnetz,
- Performanz-Wirkungen

2.1 Anforderungen an die Rechnerleistungsfähigkeit

Mit der Umsetzung des Client/Server-Konzeptes wird es nicht zu einer Reduktion bestehender Anforderungen an die Leistungsfähigkeit der Hardware-Umgebung kommen, sondern zunächst zu einer Umverteilung auf die verschiedenen Rechnersysteme. Die modulare Struktur der IuK-Systeme bestehend aus Daten-, Anwendungs- und Präsentationsschicht und deren Verteilung auf Client- und Server-Prozesse führt tendenziell sogar zu einer Intensivierung der Anforderungen. So zeigt Schill⁹ bezüglich der physischen Verteilung der Module in heterogenen Hardware-Umgebungen auf, daß die Anzahl der zum Einsatz kommenden Kommunikationsprotokolle steigen wird. Die Software-technische Abwicklung dieser Kommunikationsprotokolle erfordert neben den komplexen Daten- und Anwendungsdiensten erhöhte Prozessorleistungen von den be-

⁹ Vgl. Schill, A.: Die globale Interoperabilität ist das Ziel, in: Computerwoche Extra, Nr. 4 vom 17.09.1993, S. 13.

teiligten Rechnern. Nur durch eine effiziente Software-technische Abwicklung kann verhindert werden, daß selbst Hochleistungsnetze zu einem Engpaßfaktor werden.

Zudem werden auf Client/Server-Systeme neue Anforderungen hinzu kommen, die durch die konventionellen zentralen Hardware-Systeme nicht erbracht werden mußten. Dazu zählen unter anderem grafische Oberflächen und besonders Multi-Media-Anwendungen. Buck-Emden¹⁰ geht zum Beispiel davon aus, daß grafische Oberflächen bis zu 50 Prozent der Prozessorleistung der Front-End-Rechner für sich beanspruchen.

Aufgrund der technischen Leistungsdaten heute bereits verfügbarer Rechner-Systeme und den jährlichen Wachstumsraten von über 50 Prozent im Bereich ihrer Prozessorleistung wird der steigende Bedarf an Verarbeitungskapazitäten nicht als Problem erachtet. Man kann sogar heute davon ausgehen, daß die zur Zeit im Einsatz befindlichen Prozessoren die Grenze ihrer technischen Möglichkeiten lange noch nicht erreicht haben, da die abzuarbeitende Software oft nicht die spezifische maximale Prozessorleistungsfähigkeit beansprucht. Die Umsetzung des Client/Server-Konzeptes bietet somit die Chance, die bisher oftmals brachliegenden Möglichkeiten der 32- beziehungsweise 64-Bit-Hardware-Architekturen moderner Rechner nutzbringend einzusetzen.

2.2 Anforderungen an das Kommunikationsnetz

Bei allen Verteilungsmodellen wird die **Problematik der Netzbelastung** durch die einzelnen Verteilungsformen immer wieder aufgeworfen. Mit Hilfe bestimmter Mechanismen wie z. B. den **Stored Procedures** wird versucht, die Netzbelastung zu verringern. Wiederkehrende SQL-Abfragen lassen sich damit in der Datenbank ablegen und kommunikationsreduziert nutzen. Vom Client-Prozeß wird nur noch der Name der Stored Procedure aufgerufen. Diese kommt im Server-Teil zur Ausführung und übermittelt lediglich die Abfrageergebnisse via Netz an den Client-Prozeß zurück.

Auch wenn über solche Mechanismen versucht wird, den Kommunikationsbedarf über das Netz zu reduzieren, so herrscht dennoch ein Konsens darüber, daß Client/Server-Systeme insgesamt (im Vergleich zu monolithisch-zentralen Systemen) zu einer **signifikanten Mehrbelastung der Kommunikationswege** führen und daher sehr leistungsfähige Netze benötigt werden. Die durch Glasfaser-Technik erzielbaren Leistungsdaten im LAN-Bereich werden im allgemeinen als befriedigend eingestuft, während allerdings die zur Zeit noch verfügbaren Geschwindigkeiten im WAN-Bereich nicht ausreichend sind. Hier wird mit Breitband-ISDN und Hochgeschwindigkeitsnetzen die erforderliche Leistungsfähigkeit angestrebt.

¹⁰ Vgl. Buck-Emden, R.: Chancen für kommerzielle Lösungen in neuer Qualität - Kürzere Antwortzeiten und erweiterte Dialogmöglichkeiten steigern den Nutzen, in: Supplement zur ComputerWorld Expo, Computerwoche, Jg. 20, Nr. 43 vom 22.10.1993, S. 8.

2.3 Performanz-Wirkungen

Die Realisierung eines Client/Server-Konzeptes stellt nicht nur erhöhte Anforderungen an die Komponenten Rechner und Netzwerk. Die Verteilung der Client- und Server-Prozesse in einem Client/Server-System wird auch ausschlaggebend für die Performanz des Client/Server-Systems sein. Bei ihrer Betrachtung dieses Sachzusammenhangs lassen sich die Performanz aus Sicht des Arbeitsplatzes und aus Sicht des Gesamtsystems unterscheiden

Aus **Arbeitsplatzsicht** besteht zwischen Client- und Server-Prozeß besteht immer nur eine **temporäre, auftragstypische Beziehung**. Nach Erbringung des Dienstes seitens des Servers sind die beiden Prozesse wieder unabhängig. Der Client-Prozeß kann mit der Bearbeitung fortfahren und dabei einen weiteren Dienst entweder vom selben oder einem anderen Server-Prozeß anfragen. Sind die Prozesse physisch auf verschiedene Rechnern verteilt, bedeutet dies, daß jeweils erneut eine Kommunikationsverbindung über das Netz aufgebaut werden muß. Diese **ressourcenverbrauchende Netzoperation** ist in monolithischen Programmsystemen auf einem Zentralrechner nicht in diesem Ausmaß erforderlich. Theoretisch betrachtet müßte daher die Performanz aus Arbeitsplatzsicht aufgrund der durch die Verteilung notwendigen Netzoperationen schlechter sein. Auf der anderen Seite kann aber bei verteilten Systemen eine für die jeweiligen Prozesse **optimierte Hardware** zum Einsatz kommen, die bezüglich der Abarbeitung dieser Prozesse einen zentralen Rechner übertrifft. Inwieweit der zusätzliche Zeitaufwand der Netzoperationen durch eine Spezialisierung der Hardware-Umgebung kompensiert werden kann, hängt im konkreten Fall von einer effizienten Besetzung der Prozesse mit zugeschnittenen Hardware-Komponenten ab.

Deutlich im Vorteil sind nach Petzold und Schmitt¹¹ verteilte Strukturen aus **Sicht des Gesamtverbundes**. Sie behaupten, daß es laut empirischer Studien hier zu einer geringeren Verschlechterung der Performanz bei zunehmender Last als in monolithischen Systemen komme. Dies kann mit einer besseren Verteilung der Anwendungslast auf die verschiedenen Rechner und dem damit verbundenen späteren Erreichen der kritischen Prozessorauslastung der einzelnen Maschinen erklärt werden.

3 Migrations-Aspekte

Vor dem Hintergrund des hohen Durchdringungsgrades heutiger Unternehmen mit IuK-Systemen kann bei der Entwicklung und Einführung von Client/Server-Systemen in Unternehmen nicht von einer DV-technischen "grünen Wiese" ausgegangen werden. Ebenso wenig besteht aufgrund finanzieller, organisatorischer und kapazitiver Restriktionen die Möglichkeit, alle vorhandenen IuK-Systeme in einem Großprojekt zu einem bestimmten Stichtag auszutauschen. Es bleibt somit nur der Weg einer **aktiven Migration** von den zur Zeit existenten Systemen hin zu Client/Server-Systemen.

¹¹ Vgl. Petzold, H. J.; Schmitt, H.-J.: Verteilte Anwendungen auf der Basis von Client/Server-Architekturen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Heft Nr. 170, 1993, S. 90.

Migration wird hier als ein mittel- bis langfristig zu planender evolutionärer Prozeß gesehen, der bei der Geschäftsprozeßorientierung und -organisation der Unternehmen berücksichtigt werden muß. Das bedeutet, daß eine strategische Reorganisation der Unternehmen von einer strategischen Gesamtplanung des IuK-Bebauungsplanes flankiert sein muß, und diese beiden Aktivitäten nicht separiert werden dürfen.

Ansatzpunkte und Problemfelder der Migration sind:

- Migration der Hardware und
- Migration der Software

Da diese Aspekte stark interdependent sind und gemeinsam geplant werden müssen, ist diese Trennlinie hier rein analytischer Natur.

3.1 Migration der Hardware

Die Umsetzung des Client/Server-Konzeptes geht Hardware-technisch in der Regel mit einem **Rightsizing**, d. h., dem Anpassen der Hardware-Infrastruktur an die konkrete Unternehmenssituation einher. Aufgrund der hohen Anforderungen seitens der Client/Server-Systeme an die Hardware-Umgebung und aufgrund finanzieller Restriktionen wird das Client/Server-Konzept aber zumindest mittelfristig auf Basis existenter Drei-Ebenen-Konzepte der Unternehmen seine Realisierung finden.

Das **Drei-Ebenen-Konzept** besteht aus der Zentralen Ebene, der Abteilungsebene und der Arbeitsplatzebene und findet seine technische Begründung in der Ausnutzung spezifischer Vorteile bestimmter Rechnertypen. Als **Schnittstelle zum Anwender** haben sich im allgemeinen **PCs oder Workstations** durchgesetzt und die "dummen Terminals" der zentralistischen Lösungen verdrängt. Die Stärken dieser Arbeitsplatz-Rechner liegen in einem vorteilhaften Preis-/Leistungsverhältnis, einer flexiblen Einsatzfähigkeit und ihrem Potential, moderne Ergonomieanforderungen zu erfüllen. Für die Aufbereitung und Präsentation von Anwendungen innerhalb grafischer Oberflächen ist ein Prozessor "vor Ort" als Front-End unerlässlich, dessen Rechenleistung zusätzlich für die Individuelle Datenverarbeitung (IDV) am Arbeitsplatz zur Verfügung steht. Die auf **Abteilungsebene** platzierten **Midrange-Systeme** erfüllen im Drei-Ebenen-Konzept aufgrund ihrer Leistungsfähigkeit im Multiusing- und Multitasking-Bereich eine (meist) räumlich begrenzte Dienstleistungsfunktion für die Rechner der Arbeitsplatz-Ebene. Die mittlere Datentechnik stellt hier durch eine festgelegte Selektion und Bündelung von DV-Leistung die Verbindung zwischen der Arbeitsplatz-Ebene und den **Hintergrundsystemen der Zentralen Ebene** her. Die dort positionierten Mainframes haben ihre Stärken zum Beispiel in der Massendatenhaltung und der Datensicherung, allgemein ausgedrückt, bei den Aufgaben bereichsübergreifender Integration.

Die Rollenverteilung des Drei-Ebenen-Konzeptes steht derjenigen eines Client/Server-Konzeptes in keiner Weise entgegen. Auch und gerade im Client/Server-Konzept werden Rechnertypen unterschiedlicher Leistungsfähigkeit und **Aufgabenspezialisierung** erforderlich. Großrechner-Systeme werden weiterhin ihre Existenzberechtigung inner-

halb der Datenschicht des Client/Server-Konzeptes, zum Beispiel im Hinblick auf Integritäts- und Konsistenzwahrung, finden. Hochleistungsfähige Workstations und Mehrplatzsysteme werden die spezifischen Aufgaben dezentraler Applikations-Server erfüllen können. Die Workstations und Personal Computer der Arbeitsplatz-Ebene kommen der Flexibilisierung und Dedizierungstendenz des Client/Server-Konzeptes auf der breiten Front der Präsentationsschicht entgegen.

Krcmar¹² sieht in der Betonung dieser Rollenverteilung durch das Client/Server-Konzept eine **Renaissance des Verbundes von Spezialisten**, der zu Zeiten der General Purpose Computer zurückgedrängt wurde. Ein solcher Verbund steht in keinem strukturellen Widerspruch zu einer vom Client/Server-Konzept geforderten Hardware-Infrastruktur. Gemäß der Hinwendung zu einem logischen, Software-orientierten Client/Server-Konzept sind jedoch die Positionen und Aufgaben der Architekturelemente neu zu definieren und zu gewichten. Dabei wird die existente Hardware-Infrastruktur sukzessive ausgebaut und partiell ersetzt: eine pragmatische Vorgehensweise, um zu einer der Unternehmenssituation angepaßten Client/Server-Infrastruktur zu gelangen.

3.2 Migration der Software

Unternehmen müssen auf existente Software-Altsysteme, die auch als **”Erblastanwendungen”** oder ”Legacy Systems” bezeichnet werden, Rücksicht nehmen. Diese Legacy Systems stellen oft Investitionen in Millionenhöhe dar und können nicht schlagartig abgelöst werden. Für eine Migration sind zwei Wege vorstellbar:

Erstens kann eine schrittweise Migration zu Client/Server-Systemen durch **Modularisierung** der bestehenden monolithischen Software-Systeme gewählt werden. Bei den schon existenten Programmen wird dazu im Rahmen des Reverse-Engineerings versucht, die Struktur der sehr oft schlecht dokumentierten, nur selten klar strukturierten, gewachsenen Software zu erschließen und dann durch Software-Reengineering die Programme zeitgemäß zu modularisieren.

Zweitens ist aber auch ein sukzessiver Ersatz ganzer Teil-Systeme denkbar. In einer **Parallelentwicklung** werden auf Basis des Client/Server-Konzeptes neue Anwendungs-Teil-Systeme parallel zum Betrieb der Alt-Systeme entwickelt, getestet und zu einem Stichtag komplett ausgetauscht.

Welcher dieser beiden Migrationswege zu beschreiten ist, beziehungsweise in welchem Verhältnis die Mischung zwischen beiden Wegen ausfällt, wird maßgeblich durch den Modularisierungsgrad der vorhandenen Software, deren Integration im Gesamtsystem, ihrer operativen Bedeutung und vor allem von Risiko- und Wirtschaftlichkeitsüberlegungen getragen sein. Oftmals wird als erster Schritt einer Migration die Bereitstellung einer **einheitlichen grafischen Oberfläche** für alle Anwendungen vorgeschlagen. Dies

¹² Vgl. Krcmar, H.: Informationsmanagement und Informationssystem-Architekturen - Vorteile und Risiken von Client/Server-Architekturen aus der Sicht des Informationsmanagements, in: Krcmar, H.; Strasburger, H., (Hrsg.), Client/Server-Architekturen - Herausforderung an das Informationsmanagement, 1. Auflage, AIT Angewandte Informationstechnik, Hallbergmoos 1993, S. 16.

kann in Form der Verteilten oder Entfernten Präsentation geschehen und erscheint dann sinnvoll, wenn zum Beispiel eine monolithische Anwendung, die den Benutzern als Schnittstelle nur "dumme Terminals" zur Verfügung stellt, noch über einen längeren Zeitraum genutzt werden soll. Voraussetzung für die Ausstattung dieser Altanwendung mit einer grafischen Oberfläche sind residente Umsetzerprogramme, die die Bildschirm-inhalte einer Host-Emulations-Software als einzelne Feldinhalte transparent abfangen, um sie einer grafischen Oberfläche zu übergeben. Ein weiterer Schritt in Richtung Client/Server-Systeme ist die vollständige Auslagerung der Präsentationsschicht aus dem monolithischen Altsystem.

Anschließend kann die Auslagerung der Datenschicht in Form der **Entfernten Datenbank** erfolgen. Auch dieses Vorhaben ist bei Altanwendungen selten unproblematisch. Vielfach werden hier die Daten nicht in modernen Datenbanken gehalten, sondern in konventionellen Datei-Systemen. Abhängig vom Stand der Datenmodellierung im Unternehmen sollten Teil-Systeme identifiziert und mit Teildatenbanken hinterlegt werden. Die berechtigten Zweifel an der Realisierung eines umfassenden Unternehmensdatenmodells machen den zum Betrieb der Altsysteme parallelen Ausbau eines modularisierten Datenmodells wirtschaftlich und DV-technisch um so sinnvoller, je stärker der angestrebte Verteilungsgrad im logischen Client/Server-Konzept ausgeprägt ist.

Mit der Konzipierung des verteilten Datenmodellschemas geht der **sukzessive Ersatz monolithischer Anwendungs-Systeme** durch kooperative Anwendungen einher. Während sich Präsentations- und Datenschicht auf dem oben geschilderten Weg auch unternehmensindividuell durch standardisierte Produkte verwirklichen lassen, wird die Bereitstellung kooperativer Anwendungsmodule wegen ihrer Abhängigkeit von individuellen Geschäftsprozessen in den Unternehmen vergleichsweise aufwendiger sein. Die Entscheidungssituation "Make-or-Buy" beeinflusst hier maßgeblich den Migrationsweg und wirft die bekannten Fragen der Herstellerbindung und vor allem der quantitativen und qualitativen Funktionsabdeckung auf.

Der oben geschilderte Weg kann nur als Beispiel dienen. Ein generelles Migrations-Rezept kann hier nicht vorgeschlagen werden, da die individuelle Unternehmenssituation ausschlaggebend für den Aufsattpunkt und die Ausgestaltung des Migrationsweges ist. Grundsätzlich ermöglicht jedoch das sukzessive Vorgehen eine Begrenzung des mit der Einführung von Client/Server-Systemen verbundenen Risikopotentials.

Literaturverzeichnis

- Buck-Emden, R.: Chancen für kommerzielle Lösungen in neuer Qualität - Kürzere Antwortzeiten und erweiterte Dialogmöglichkeiten steigern den Nutzen, in: Supplement zur ComputerWorld Expo, Computerwoche, Jg. 20, Nr. 43 vom 22.10.1993, S. 8-9.
- Bues, M.: Offene Systeme - Strategien, Konzepte und Techniken für das Informationsmanagement, Springer-Verlag, Berlin u. a. 1994.
- Hansen, W.-R.: Client/Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, in: Hansen, W.-R. (Hrsg.), Client/Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, Addison-Wesley Publishing Company, Bonn u. a. 1993, S. 13-67.
- Kargl, H., Controlling im DV-Bereich, 2. Aufl., Oldenbourg-Verlag, München, Wien 1994.
- Krcmar, H.: Informationsmanagement und Informationssystem-Architekturen - Vorteile und Risiken von Client/Server-Architekturen aus der Sicht des Informationsmanagements, in: Krcmar, H.; Strasburger, H., (Hrsg.), Client/Server-Architekturen - Herausforderung an das Informationsmanagement, 1. Auflage, AIT Angewandte Informationstechnik, Hallbergmoos 1993, S. 9-29.
- Müller, B.; Starke, Th.: CASE für Client/Server ist mehr als CASE - Eine OLTP-Anwendung für das Kraftwerks-Service-Geschäft bei ABB, in: Computerwoche Extra, Nr. 4 vom 17.09.1993, S. 28-36.
- Petzold, H. J.; Schmitt, H.-J.: Organisatorische Gestaltungsmöglichkeiten des Einsatzes von Client/Server-Systemen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Heft Nr. 174, 1993, S. 73-87.
- Petzold, H. J.; Schmitt, H.-J.: Verteilte Anwendungen auf der Basis von Client/Server-Architekturen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Heft Nr. 170, 1993, S. 79-92.
- Renaud, P. E.: Introduction To Client/Server Systems - A Practical Guide for Systems Professionals, John Wiley & Sons Inc., New York u. a. 1993.
- Schill, A.: Die globale Interoperabilität ist das Ziel, in: Computerwoche Extra, Nr. 4 vom 17.09.1993, S. 12-15, 45.

Bisher erschienen

Stand: Dezember 2000 – Den aktuellen Stand der Reihe erfahren
Sie über unsere Web Site unter <http://wi.uni-giessen.de>

Nr. 1/1996	Grundlagen des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 2/1996	Wettbewerbs- und Organisationsrelevanz des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 3/1996	Realisierungsaspekte des Client/Server-Konzepts	Schwicker/Grimbs
Nr. 4/1996	Der Geschäftsprozeß als formaler Prozeß - Definition, Eigenschaften, Arten	Schwicker/Fischer
Nr. 5/1996	Manuelle und elektronische Vorgangsteuerung.....	Schwicker/Rey
Nr. 6/1996	Das Internet im Unternehmen - Neue Chancen und Risiken	Schwicker/Ramp
Nr. 7/1996	HTML und Java im World Wide Web.....	Gröning/Schwicker
Nr. 8/1996	Electronic-Payment-Systeme im Internet.....	Schwicker/Franke
Nr. 9/1996	Von der Prozeßorientierung zum Workflow-Management - Teil 1: Grundgedanken, Kernelemente, Kritik	Maurer
Nr. 10/1996	Von der Prozeßorientierung zum Workflow- Management - Teil 2: Prozeßmanagement und Workflow	Maurer
Nr. 11/1996	Informationelle Unhygiene im Internet.....	Schwicker/Dietrich/Klein
Nr. 12/1996	Towards the theory of Virtual Organisations: A description of their formation and figure.....	Appel/Behr
Nr. 1/1997	Der Wandel von der DV-Abteilung zum IT-Profitcenter: Mehr als eine Umorganisation.....	Kargl
Nr. 2/1997	Der Online-Markt - Abgrenzung, Bestandteile, Kenngrößen	Schwicker/Pörtner
Nr. 3/1997	Netzwerkmanagement, OSI Framework und Internet SNMP	Klein/Schwicker
Nr. 4/1997	Künstliche Neuronale Netze - Einordnung, Klassifikation und Abgrenzung aus betriebswirtschaftlicher Sicht	Strecker/Schwicker
Nr. 5/1997	Sachzielintegration bei Prozeßgestaltungsmaßnahmen.....	Delnef
Nr. 6/1997	HTML, Java, ActiveX - Strukturen und Zusammenhänge.....	Schwicker/Dandl
Nr. 7/1997	Lotus Notes als Plattform für die Informationsversorgung von Beratungsunternehmen.....	Appel/Schwaab
Nr. 8/1997	Web Site Engineering - Modelltheoretische und methodische Erfahrungen aus der Praxis	Schwicker
Nr. 9/1997	Kritische Anmerkungen zur Prozeßorientierung	Maurer/Schwicker
Nr. 10/1997	Künstliche Neuronale Netze - Aufbau und Funktionsweise	Strecker
Nr. 11/1997	Workflow-Management-Systeme in virtuellen Unternehmen	Maurer/Schramke
Nr. 12/1997	CORBA-basierte Workflow-Architekturen - Die objektorientierte Kernanwendung der Bausparkasse Mainz AG	Maurer
Nr. 1/1998	Ökonomische Analyse Elektronischer Märkte.....	Steyer
Nr. 2/1998	Demokratiopolitische Potentiale des Internet in Deutschland	Muzic/Schwicker
Nr. 3/1998	Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil 1)	Delnef
Nr. 4/1998	Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil 2)	Delnef
Nr. 5/1998	Betriebswirtschaftlich-organisatorische Aspekte der Telearbeit	Polak
Nr. 6/1998	Das Controlling des Outsourcings von IV-Leistungen	Jäger-Goy
Nr. 7/1998	Eine kritische Beurteilung des Outsourcings von IV-Leistungen.....	Jäger-Goy
Nr. 8/1998	Online-Monitoring - Gewinnung und Verwertung von Online-Daten.....	Guba/Gebert
Nr. 9/1998	GUI - Graphical User Interface.....	Maul
Nr. 10/1998	Institutionenökonomische Grundlagen und Implikationen für Electronic Business.....	Schwicker
Nr. 11/1998	Zur Charakterisierung des Konstrukts "Web Site".....	Schwicker
Nr. 12/1998	Web Site Engineering - Ein Komponentenmodell.....	Schwicker
Nr. 1/1999	Requirements Engineering im Web Site Engineering – Einordnung und Grundlagen.....	Schwicker/Wild
Nr. 2/1999	Electronic Commerce auf lokalen Märkten	Schwicker/Lüders
Nr. 3/1999	Intranet-basiertes Workgroup Computing	Kunow/Schwicker
Nr. 4/1999	Web-Portale: Stand und Entwicklungstendenzen.....	Schumacher/Schwicker
Nr. 5/1999	Web Site Security.....	Schwicker/Häusler
Nr. 6/1999	Wissensmanagement - Grundlagen und IT-Instrumentarium.....	Gaßen
Nr. 7/1999	Web Site Controlling.....	Schwicker/Beiser
Nr. 8/1999	Web Site Promotion	Schwicker/Arnold
Nr. 9/1999	Dokumenten-Management-Systeme – Eine Einführung	Dandl
Nr. 10/1999	Sicherheit von eBusiness-Anwendungen – Eine Fallstudie	Harper/Schwicker
Nr. 11/1999	Innovative Führungsinstrumente für die Informationsverarbeitung	Jäger-Goy
Nr. 12/1999	Objektorientierte Prozeßmodellierung mit der UML und EPK	Dandl
Nr. 1/2000	Total Cost of Ownership (TCO) – Ein Überblick.....	Wild/Herges
Nr. 2/2000	Implikationen des Einsatzes der eXtensible Markup Language – Teil 1: XML-Grundlagen.....	Franke/Sulzbach
Nr. 3/2000	Implikationen des Einsatzes der eXtensible Markup Language – Teil 2: Der Einsatz im Unternehmen	Franke/Sulzbach
Nr. 4/2000	Web-Site-spezifisches Requirements Engineering – Ein Formalisierungsansatz	Wild/Schwicker
Nr. 5/2000	Elektronische Marktplätze – Formen, Beteiligte, Zutrittsbarrieren	Schwicker/Pfeiffer
Nr. 6/2000	Web Site Monitoring – Teil 1: Einordnung, Handlungsebenen, Adressaten.....	Schwicker/Wendt
Nr. 7/2000	Web Site Monitoring – Teil 2: Datenquellen, Web-Logfile-Analyse, Logfile-Analyzer	Schwicker/Wendt
Nr. 8/2000	Controlling-Kennzahlen für Web Sites.....	Schwicker/Wendt
Nr. 9/2000	eUniversity – Web-Site-Generierung und Content Management für Hochschuleinrichtungen.....	Schwicker/Ostheimer/Franke

Bestellung (bitte kopieren, ausfüllen, zusenden/zufaxen)

Adressat: Professur für BWL und Wirtschaftsinformatik
 Fachbereich Wirtschaftswissenschaften
 Licher Straße 70
 D – 35394 Gießen
 Telefax: (0 641) 99-22619

Hiermit bestelle ich gegen Rechnung die angegebenen Arbeitspapiere zu einem Kostenbeitrag von DM 10,- pro Exemplar (MwSt. entfällt) zzgl. DM 5,- Versandkosten pro Sendung.

Nr.	An
1/1996	
2/1996	
3/1996	
4/1996	
5/1996	
6/1996	
7/1996	
8/1996	
9/1996	
10/1996	
11/1996	
12/1996	

Nr.	An
1/1997	
2/1997	
3/1997	
4/1997	
5/1997	
6/1997	
7/1997	
8/1997	
9/1997	
10/1997	
11/1997	
12/1997	

Nr.	Anz
1/1998	
2/1998	
3/1998	
4/1998	
5/1998	
6/1998	
7/1998	
8/1998	
9/1998	
10/1998	
11/1998	
12/1998	

Nr.	Anz
1/1999	
2/1999	
3/1999	
4/1999	
5/1999	
6/1999	
7/1999	
8/1999	
9/1999	
10/1999	
11/1999	
12/1999	

Nr.	Anz
1/2000	
2/2000	
3/2000	
4/2000	
5/2000	
6/2000	
7/2000	
8/2000	
9/2000	

Absender:

Organisation _____

Abteilung _____

Nachname, Vorname _____

Straße _____

Plz/Ort _____

Telefon _____ Telefax _____ eMail _____

Ort, Datum _____ Unterschrift _____