

Konfliktbehandlung in komponentenorientierten betrieblichen Anwendungssystemen

Klement J. Fellner

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Institut für Technische und Betriebliche Informationssysteme
Wirtschaftsinformatik
Postfach 41 20, 39016 Magdeburg, Deutschland
Telefon +49 391 67-183 86, Fax -112 16
Email: fellner@iti.cs.uni-magdeburg.de
URL: <http://www-wi.cs.uni-magdeburg.de>

Abstract

Ausgehend von der Definition des Begriffs Fachkomponente (für betriebliche Anwendungsdomänen nutzbare Komponenten) und elementare Fachkomponente (aus Sicht der Anwendungsdomäne kleinstmögliche Fachkomponente) werden am Beispiel der Erweiterung von PPS-Systemen um eine Recyclingkomponente mögliche Konflikte auf semantischer Ebene aufgezeigt und ein Lösungsansatz durch eine „vermittelnde Komponente“ – dem Linkobjekt – beschrieben.

1 Motivation und Begriffsdefinition

Betriebliche Informationssysteme sind – mehr denn je – das Rückgrat der betrieblichen Leistungserstellung und mitverantwortlich für Erfolg oder Mißerfolg unternehmerischer Tätigkeit (vgl. Alpar et al. 1998, S. 4ff.). Im Zuge zunehmender Konkurrenz müssen Unternehmen ihre Geschäftsprozesse effizienter und effektiver umsetzen, um dadurch Wettbewerbsvorteile zu erzielen. Neben der Verbesserung der Geschäftsprozesse kommt der Verfügbarkeit aktueller Information über inner- sowie zwischenbetrieblicher Leistungsgrößen (z. B. Durchlaufzeiten, Lieferbereitschaft, etc.) und somit der informationstechnischen Unterstützung eine besondere Rolle zu. Die Unterstützung durch Informationssysteme muß dabei sowohl effizient, d.h. die Kostenersparnis muß höher sein als der Aufwand für die Einführung des Systems, als auch effektiv im Sinne von meßbaren Prozeßverbesserungen (z. B. Verringerung der Durchlaufzeit) sein.

Unternehmen haben bei der Umsetzung in konkrete Anwendungssysteme die Wahl zwischen Eigenentwicklung der Informationssysteme, dem Kauf von (heterogenen) Speziallösungen für einzelne Unternehmensbereiche oder der Anschaffung und Anpassung einer integrierten Standardlösung. Eigenentwicklungen haben bei erfolgreicher Umsetzung gegenüber Standardlösungen den Vorteil einer effektiveren Unterstützung der spezifischen Geschäftsprozesse eines Unternehmens. Nachteilig wirken sich bei einer Neuentwicklung die

höheren Risiken in Bezug auf Kapital- und Personaleinsatz aus. Dabei darf nicht außer Acht gelassen werden, daß die Einführung (inkl. Anpassung) einer Standardlösung gegenüber einer Individualentwicklung nicht unbedingt „günstiger“ ist. Dies und die Notwendigkeit einer weitreichenden und aufwendigen Anpassung von Standardlösungen an das spezifische Umfeld eines Unternehmens führt gegenüber dem bislang vorherrschendem „make or buy“ zu einem „make and buy“ (vgl. Kurbel et al. 1994). Dabei werden Spezialsysteme für die Unterstützung der Kernprozesse eines Unternehmens herangezogen und Prozesse, die durch Gesetze o. ä. vorgegeben sind (z. B. Rechnungswesen, Personal, ...), mittels angepasster Standardsoftware abgedeckt.

Die branchen- oder fachspezifischen Spezialsysteme können nun individuell entwickelt oder am Markt eingekauft werden. Der Vorteil der optimalen Prozessunterstützung, der durch den Einsatz von Spezialsystemen entsteht, wird jedoch durch den notwendigen Integrationsaufwand geschmälert, welcher notwendig ist, um einen Datenaustausch zwischen zumeist *heterogenen* Systemen zu ermöglichen.

Vor allem das Fehlen von *offenen* und *standardisierten* Schnittstellen auf *fachlicher* Ebene verhindert eine mühelose Integration unterschiedlicher Systeme in ein Gesamtsystem. Gremien, wie z. B. die Object Management Group (OMG) oder die Open Application Group (OAG) versuchen über eine Standardisierung von Geschäftsobjekten (vgl. OMG 1996) bzw. eines standardisierten Datenaustauschformats (vgl. OAG 1997) eine standardisierte Kommunikation zwischen unterschiedlichen Systemen zu ermöglichen. Beiden Standardisierungsvorschlägen ist neben dem *Business Object* eine *komponentenorientierte* Softwarearchitektur gemeinsam.

Das Ziel aller Bestrebungen im Umfeld der komponentenorientierten Softwareentwicklung ist neben der Wiederverwendung einzelner Softwarebausteine ein einfaches Austauschen veralteter bzw. Hinzufügen neuer Komponenten – ein „*plug & play*“ von Komponenten.

Aufbauend auf die in der Literatur vorherrschenden Definitionen wird eine Komponente wie folgt definiert (für eine Diskussion der verschiedenen Ansätze siehe Szyperski 1998, S. 30-34):

Eine Komponente ist ein *wiederverwendbares, abgeschlossenes* und *vermarktbares* Stück Software, das über eine *wohldefinierte Schnittstelle* Dienste anbietet und in zur Zeit der Entwicklung *unvorhersehbaren Kombinationen* mit anderen Fachkomponenten eingesetzt und an unternehmensspezifische Erfordernisse angepaßt werden kann.

An Stelle des Begriffs „Komponente“ wird für Komponenten, die Teil eines betrieblichen Anwendungssystems sind, der enger gefaßte Begriff der „Fachkomponente“ (Fellner et al. 1999) verwendet.

Eine *Fachkomponente* ist eine Komponente, die eine bestimmte Menge von Diensten einer *betrieblichen* Anwendungsdomäne implementiert.

2 Elementare Fachkomponenten und Linkobjekte zur Lösung fachlicher Konflikte

Der Definition folgend stellt eine Fachkomponente für einen Anwendungsbereich eine Menge von Funktionen über entsprechende Dienste zur Verfügung. Der Kunde wählt nun die für seine Anforderungen passenden Fachkomponenten aus und erstellt damit sein individuelles Anwendungssystem. Werden die vom Kunden ausgewählten Fachkomponenten zusammengefügt, können prinzipiell zwei Fälle unterschieden werden:

- Die Fachkomponenten stellen (zum Teil) gleiche Dienste zur Verfügung, so daß ein *fachlicher Konflikt* entsteht oder
- die Fachkomponenten sind bezüglich der angebotenen Dienste disjunkt.

Ein *fachlicher Konflikt* entsteht dann, wenn eine oder mehrere verschiedene Fachkomponenten, die (logisch) in das selbe Anwendungssystem integriert sind, die selben Dienste einer bestimmten Anwendungsdomäne anbieten. Jeweils zwei Fachkomponenten werden (bezüglich der angebotenen Dienste) als *disjunkt* bezeichnet, wenn kein Dienst einer bestimmten Anwendungsdomäne existiert, der von beiden Fachkomponenten implementiert wird. In Abgrenzung zu fachlichen Konflikten beziehen sich *technische Konflikte* auf Probleme, die mit den DV-Technik-nahen Aspekten eines komponentenorientierten betrieblichen Anwendungssystems im Zusammenhang stehen. Dazu gehören beispielsweise Probleme der Adressierung von (entfernten) Funktionen oder Methoden, die gemeinsame Nutzung von Speicherbereichen oder Betriebsmitteln sowie das Einhalten bestimmter Kommunikationsprotokolle. Diese Probleme werden in diesem Beitrag nicht aufgegriffen.

Fachliche Konflikte sind prinzipiell durch zwei Maßnahmen lösbar:

- Durch das ausschließliche Zulassen von disjunkten Fachkomponenten zur Integration in ein bestimmtes Anwendungssystem oder
- durch die Bereitstellung von Techniken zur Behebung fachlicher Konflikte.

Zur Umsetzung der ersten Maßnahme bedarf es der *Standardisierung* von Fachkomponenten, um die Entstehung fachlicher Konflikte von vornherein auszuschließen. Dabei ist zur Umsetzung der ersten Maßnahme insbesondere zu fordern, daß im Rahmen der Standardisierung ermittelte Mengen von Diensten disjunkt sind. Die Standardisierung von Fachkomponenten ermöglicht des weiteren nur durch eine damit verbundene Definition eines einheitlichen Begriffssystems¹ eine Behebung fachlicher Konflikte.

Es ist jedoch auch offensichtlich, daß hinsichtlich der Praxistauglichkeit komponentenorientierter betrieblicher Anwendungssysteme Fachkomponenten von (zumindest) mittlerer Größe zugelassen werden müssen, die insbesondere nicht disjunkt sind. Um dieses Problem zu lösen, werden folgende Grundkonzepte vorgeschlagen:

- Die Standardisierung elementarer Fachkomponenten und
- die Behebung fachlicher Konflikte mit Hilfe von Linkobjekten.

Elementare Fachkomponenten sind Gegenstand einer (feingranularen) Standardisierung. Sie sind bezüglich der angebotenen Dienste paarweise disjunkt. Die Vereinigungsmenge über alle Dienste, die von den elementaren Fachkomponenten einer bestimmten Anwendungsdomäne angeboten werden, ergibt genau die zu der Anwendungsdomäne gehörenden Dienste. Fachkomponenten werden durch die Kombination elementarer Fachkomponenten zusammengesetzt. Fachkomponenten müssen ihrerseits bezüglich der von ihnen angebotenen Dienste nicht disjunkt sein, d. h.: verschiedene Fachkomponenten dürfen als Teilmenge die gleiche Menge elementarer Fachkomponenten umfassen (vgl. Fellner et al. 1999).

Die dadurch auftretenden fachlichen Konflikte werden mittels einer vermittelnden Komponente – den *Linkobjekten* - gelöst. Linkobjekte sind spezielle Softwarebausteine, die dazu dienen Dienstanfragen und Ergebnisse von Dienstanfragen weiterzuleiten. Jede Dienstanfrage, für die ein fachlicher Konflikt besteht, wird zu diesem Zweck durch ein

¹ Vor allem für den Bereich der Übermittlung betrieblicher Daten sind bereits entsprechende Begriffssysteme definiert. Den bekanntesten Standard hierzu ist sicherlich UN/EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport) (UN 1995).

Linkobjekt vermittelt. Eine Fachkomponente muß daher in der Lage sein, für jede elementare Fachkomponente die sie umfaßt, eine Schnittstelle für die Menge von Diensten zu generieren, die der elementaren Fachkomponente zugeordnet sind. Ist eine solche Schnittstelle vorhanden, dann erfolgt jede Nutzung der Dienste über diese Schnittstelle. Das gilt sowohl für Anfragen von anderen Fachkomponenten als auch für Anfragen, die von anderen Teilen der selben Fachkomponente gestellt werden. Zur Sicherstellung einer höheren Effizienz, sollen diese Schnittstellen jedoch erst dann generiert werden, wenn sie aufgrund eines fachlichen Konflikts benötigt werden.

Abbildung 1 illustriert die Konfliktbehandlung mit Linkobjekten für ein Beispiel, bei dem ein fachlicher Konflikt bezüglich der Dienste besteht, die von der Fachkomponente L bereitgestellt werden. Dabei wird davon ausgegangen, daß ein Unternehmen (der Kunde) ein integriertes Anwendungssystem aus Fachkomponenten, die von unterschiedlichen Herstellern (A und B) angeboten werden, implementiert. Um das Beispiel einfach zu halten, wird angenommen, daß die Fachkomponente L gleichzeitig eine elementare Fachkomponente ist. Für die Fachkomponente M von Anbieter A werden nun zum Zwecke der fachlichen Konfliktbehandlung zwei Schnittstellen generiert, ein Client-Adapter (1) und ein Server-Adapter (2). Der Server-Adapter wird generiert, weil die Fachkomponente eine Menge von Diensten anbietet. Der Client-Adapter ist notwendig, da die Fachkomponente intern Dienste der Fachkomponente L nachfragt. Für diese Anfragen muß nun aufgrund des fachlichen Konflikts sichergestellt werden, daß diese für das gesamte Anwendungssystem immer von der gleichen Stelle bearbeitet werden. Wenn kein fachlicher Konflikt besteht, dann fallen Client- und Server-Adapter zu einer Schnittstelle zusammen.

Für das Weiterleiten von Anfragen nach Diensten der Fachkomponente L wird ferner ein Linkobjekt eingeführt. Das Linkobjekt besteht im wesentlichen aus einem Client- und einem Server-Adapter und einer Logik, die angibt auf welchen Server-Adapter einer Fachkomponente Dienstanfragen letztendlich weitergegeben werden sollen. In dem abgebildeten Fall werden allen Dienstanfragen an die Fachkomponente L von Anbieter B weitergegeben. Insbesondere auch Anfragen, die im Normalfall von der Fachkomponente M von Anbieter A selbst bearbeitet werden können (3). Anfragen nach Diensten der Fachkomponente M, die von anderen Teilen des integrierten Anwendungssystems kommen und die zuvor direkt von der M bearbeitet wurden (4), werden nun ebenso über das Linkobjekt geroutet (5). Ein Linkobjekt ersetzt damit systemweit alle Schnittstellen zu Diensten, die einer bestimmten konfliktären elementaren Fachkomponente zugeordnet sind.

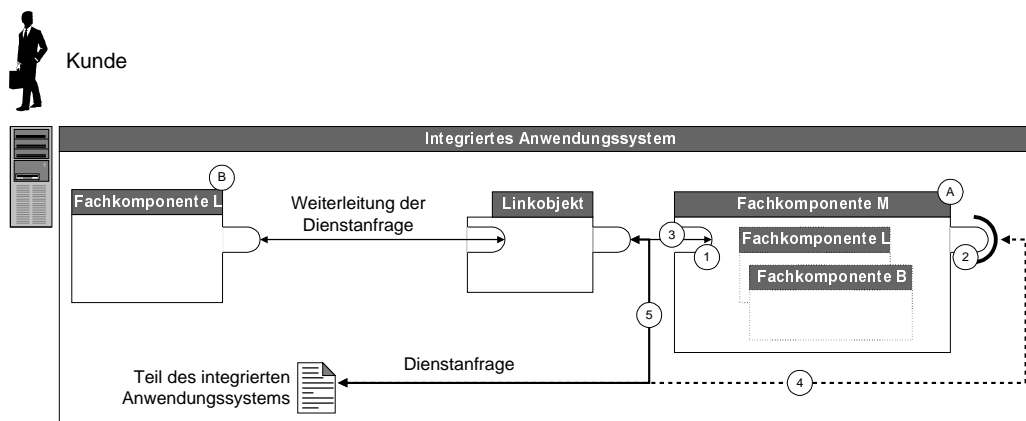


Abbildung 1: Behandlung fachlicher Konflikte mit Linkobjekten

Im folgenden Abschnitt wird gezeigt, wie dieses allgemeine Vorgehen zur Konfliktlösung zur Erweiterung der Stammdatenverwaltung eines Systems zur Produktionsplanung und -steuerung (PPS) um Recyclingfunktionalität angewendet werden kann.

3 Fachkomponenten für Produktions-, Recyclingplanungs- und steuerungssysteme

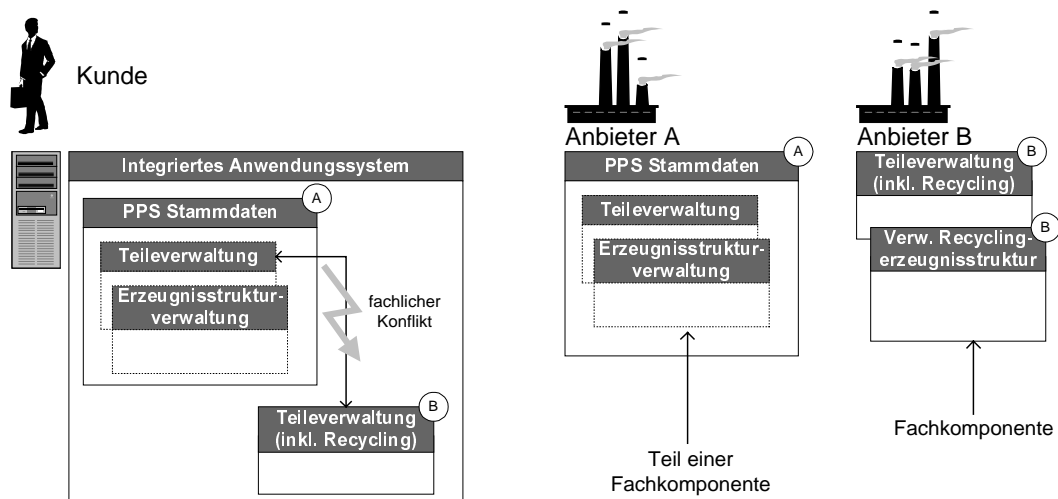


Abbildung 2: Anbieter B bietet Fachkomponenten mit zusätzlicher Funktionalität an

Für eine integrierte Planung von Produktions- sowie Recyclingprozessen müssen aktuell verfügbare PPS-Systeme um die zusätzlich benötigten Funktionen erweitert werden. Im wesentlichen sind dies Erweiterungen im Bereich der Stammdatenverwaltung und der Materialwirtschaft. In der Stammdatenverwaltung ist unter anderem eine Verwaltung von für das Recycling relevanter Teileinformationen vorzusehen. Zusätzliche Informationen stellen z. B. Recyclingmerkmale, wie *Qualitätsabweichungen*, *Lagerfähigkeit*, *Recyclinghäufigkeit* oder *Vorschriften* dar. (für eine umfassende Betrachtung der notwendigen Erweiterungen sei auf Rautenstrauch 1996, verwiesen).

Im folgenden Szenario gehen wir davon aus, daß zwei Unternehmer eine Fachkomponente zur Verwaltung von Stammdaten der PPS anbieten. Die Fachkomponenten von Anbieter B bieten dabei zusätzliche Dienste zur Verwaltung von Recyclinginformationen sowie zur Generierung von Recycling-Erzeugnisstrukturen an (siehe Abbildung 2). Will der Kunde sein bereits vorhandenes PPS-System (von Anbieter A) um Funktionalitäten zur Recyclingplanung und -steuerung erweitern, muss er zusätzlich zur Fachkomponente für die Generierung und Verwaltung der Recycling-Erzeugnisstrukturen („Verw. Recycling-Erzeugnisstruktur“) auch die erweiterte Fachkomponente „Teilverwaltung (inkl. Recycling)“ einsetzen.

In Abbildung 3 ist die Fachkomponente „PPS_Stammdaten“ mit ihren Unterkomponenten und deren Abhängigkeiten dargestellt. Die Fachkomponente „Erzeugnisstrukturverwaltung“ benötigt von der „Teilverwaltung“ Informationen über die in der Erzeugnisstruktur vorkommenden Teile, bzw. baut die Erzeugnisstruktur ausgehend von Teileinformationen auf. Diese Informationen werden von der „Teilverwaltung“ über den Dienst *b* zur Verfügung gestellt. Dienst *b* wird in diesem Fall nur lokal in der Fachkomponente „PPS_Stammdaten“ nachgefragt. Im Gegensatz zu den Diensten *a*, *x* und *y*, die von anderen Fachkomponenten nachgefragt werden und über die Fachkomponente „PPS_Stammdaten“ zur Verfügung gestellt werden.

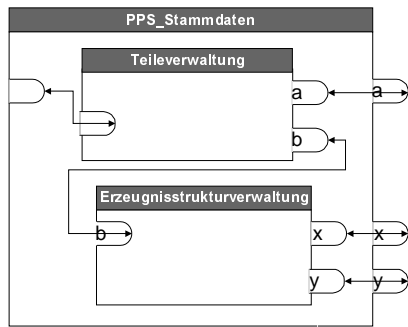


Abbildung 3: Stammdatenverwaltung ohne Recyclinginformationen

Wird nun diese Systemkonfiguration um die erweiterte Teileverwaltung und die Fachkomponente zur Verwaltung der Recycling-Erzeugnisstrukturen erweitert, kommt es zu einem Konflikt zwischen der vorhandenen „Teileverwaltung“ und der „Teileverwaltung (inkl. Recycling)“. Beide Fachkomponenten bieten den Dienst *b* (z. B. Abfrage Teilebeschreibung) an. Würde nur eine Fachkomponente den Dienst benötigen, könnte dieser Konflikt einfach gelöst werden, indem die Dienstkomponente die Dienstanfrage bei der neuen Serverkomponente stellt. Da Dienst *b* aber sowohl von der Fachkomponente „Erzeugnisstrukturverwaltung“ als auch von „Verw. Recycling-Erzeugnisstruktur“ nachgefragt wird, ist eine Weiterleitung beider Anfragen an die neue Fachkomponente notwendig. Dazu wird ein Linkobjekt eingefügt, welches jede Nachfrage nach Dienst *b* an die anbietende Fachkomponente weiterleitet. Die anbietende Fachkomponente muss sich dazu beim Linkobjekt anmelden. Stehen mehrere Fachkomponenten zur Auswahl, die einen bestimmten Dienst anbieten können, muss eine Fachkomponente ausgewählt werden, die ab diesem Zeitpunkt alle Dienstanfragen bearbeitet. In unserem Beispiel wurde die Fachkomponente „Teileverwaltung“ komplett durch die erweiterte Fachkomponente ersetzt (Dienst *a* wird nicht mehr von „PPS_Stammdaten“ für andere Fachkomponenten angeboten). Da jedes Linkobjekt für einen Dienst erzeugt wird, wäre auch eine Aufteilung der Dienstbearbeitung auf beide Fachkomponenten möglich. Die Fachkomponente „Teileverwaltung“ könnte sowohl die Dienste *a* als auch *b* weiterhin anbieten. Damit würde „Teileverwaltung (inkl. Recycling)“ nur die zusätzliche Funktionalität anbieten (Dienst *r*).

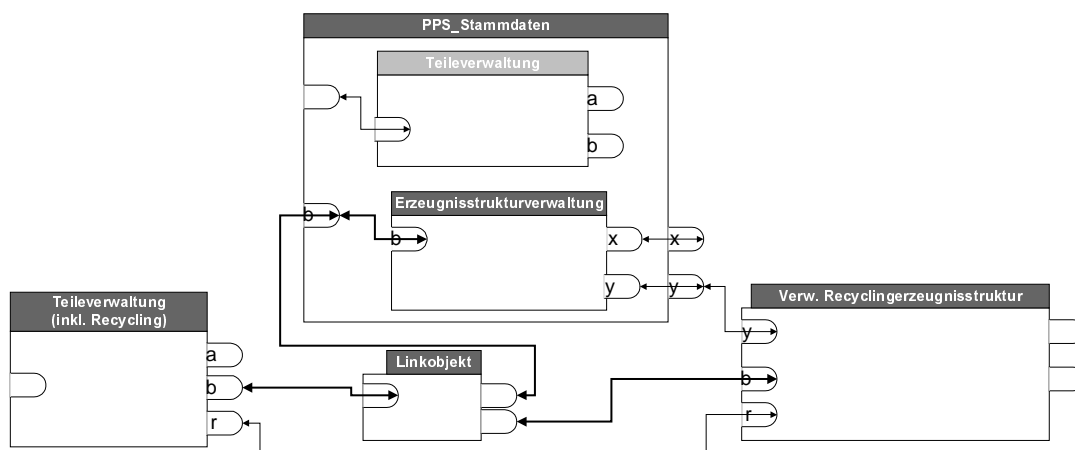


Abbildung 4: Erweiterung der Stammdatenverwaltung um Recyclinginformationen und Konfliktauflösung

4 Zusammenfassung und Ausblick

In diesem Beitrag wurde die Problematik der Konfliktauflösung bei Komposition von betrieblichen Anwendungssystemen mit Fachkomponenten aufgezeigt und anhand eines Beispiels (Erweiterung von PPS-Systemen um Funktionalität zur Recyclingplanung und -steuerung) eine Lösungsmöglichkeit dargestellt. Dazu wurde das Konzept der elementaren Fachkomponenten und der Linkobjekte eingeführt.

Aktuell wird ein bereits bestehender Forschungsprototyp eines komponentenorientierten betrieblichen Anwendungssystems um die vorgestellte Funktionalität zur Generierung bzw. Verwaltung von Recycling-Erzeugnisstrukturen erweitert.

Literatur

- Alpar, P.; Grob, H. L.; Weimann, P.; Winter, R. (1998): Unternehmensorientierte Wirtschaftsinformatik: eine Einführung in die Strategie und Realisierung erfolgreicher IuK-Systeme. Braunschweig.
- Fellner, K. J.; Rautenstrauch, C.; Turowski, K. (1999): Fachkomponenten zur Gestaltung betrieblicher Anwendungssysteme. erscheint in: Information Management & Consulting 99(2).
- Kurbel, K.; Rautenstrauch, C.; Opitz, B.; Scheuch, R. (1994): From »Make or Buy« to »Make and Buy«: Tailoring Information Systems Through Integration Engineering. Journal of Database Management 5(1994), S. 18-30.
- OAG (Hrsg.) (1997): Open Applications Integration.
<ftp://ftp.openapplications.org/openapplications.org/whtpaper.zip>. 16.06.1998.
- OMG (Hrsg.) (1996): Common Facilities RFP-4: Common Business Objects and Business Object Facility. <ftp://ftp.omg.org/pub/docs/cf/96-01-04.pdf>. 21. Jan. 1998.
- Rautenstrauch, C. (1996): Fachkonzept für ein integriertes Produktions-, Recyclingplanungs- und Steuerungssystem. Berlin.
- Szyperski, C. (1998): Component Software: Beyond Object-Oriented Programming. 2. Aufl., Harlow.
- UN (Hrsg.) (1995): United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport.
<http://www.unece.org/trade/untdid/Welcome.html>. 01.12.1998.

