

# Systematische Identifikation von Fachkomponenten mit SOM

Sebastian Schlauderer, Martin Robel, and Thomas Friedrich

University of Bamberg, Chair of Industrial Information Systems, Bamberg, Germany  
{sebastian.schlauderer, martin.robel, thomas.friedrich}@uni-bamberg.de

**Abstract.** Aufgrund der stetig wachsenden Herausforderungen ist es für Unternehmen heutzutage von großer Bedeutung, Informationssysteme flexibel und kostengünstig anpassen zu können. Hierfür ist es jedoch unabdingbar, dass die Software-Bausteine eines Systems entsprechend systematisch abgeleitet und gestaltet werden. Die Business-Component-Identification-Methode (BCI-Methode) verspricht vor dem Hintergrund der Komponentenorientierung die Identifikation geeigneter Software-Bausteine zu gewährleisten. Ein Vorteil dieser Methode ist es, dass sie unabhängig von der Art der zugrundeliegenden konzeptionellen Modelle angewandt werden kann. Da die Qualität der Modelle jedoch die Güte der gefundenen Lösung maßgeblich determiniert, ist es indes fraglich, welche Modellarten sich besonders gut eignen. Die Methodik des Semantischen Objektmodells (SOM) bietet einen ganzheitlichen und etablierten Ansatz zur systematischen Beschreibung des Fachkonzepts. Im Rahmen des Artikels wird daher untersucht, inwiefern die Kombination von BCI- und SOM-Methodik zu validen Ergebnissen führt. Hierfür wird insbesondere untersucht, welche Modelle betrachtet werden müssen und welche Beziehungen zu unterscheiden sind.

**Keywords:** Komponentenorientierung, SOM-Methodik, BCI-Methode

## 1 Motivation

Die Entwicklung von Informationssystemen steht heutzutage vor einer Vielzahl teils konterkarierender Herausforderungen. Einerseits ist es notwendig, auf sich ändernde Marktbedingungen schnell reagieren zu können und flexibel Funktionalität hinzufügen zu können [1]. Andererseits wächst der Kostendruck auf die Entwicklungsabteilungen [1]. Vor diesem Hintergrund verspricht die modulare Entwicklung von Informationssystemen eine Vielzahl an Vorteilen [2], [3]. Durch die Strukturierung von Informationssystemen in einzelne, lose gekoppelte Bausteine wird es ermöglicht, Informationssysteme flexibel anzupassen oder zu erweitern. Durch die Wiederverwendung dieser Bausteine, auch Fachkomponenten genannt [4], können weiterhin Entwicklungszeiten und Entwicklungskosten drastisch gekürzt werden [5]. Im Vergleich zur reinen Objektorientierung kann durch das Zusammenfassen mehrerer Objekte zu einer Fachkompo-

13<sup>th</sup> International Conference on Wirtschaftsinformatik,  
February 12-15, 2017, St. Gallen, Switzerland

Schlauderer, S.; Robel, M.; Friedrich, T. (2017): Systematische Identifikation von Fachkomponenten mit SOM, in Leimeister, J.M.; Brenner, W. (Hrsg.): Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017), St. Gallen, S. 316-330

nente das Problem fragiler Basisklassen eingegrenzt werden und gleichermaßen sichergestellt werden, dass Vererbungsstrukturen das Geheimnisprinzip nicht durchbrechen [6].

Aufgrund der erwarteten Vorteile wurde eine Vielzahl an Methoden und Werkzeugen sowohl in der wissenschaftlichen Literatur als auch in der Praxis vorgeschlagen, welche die Konzeption, die Implementierung sowie die Assemblierung von Fachkomponenten zum Ziel haben [2], [6-9]. Insbesondere die Frage, wie ein Informationssystem möglichst effizient in Fachkomponenten partitioniert werden kann, ist von zentraler Bedeutung, da hierdurch die Anpassbarkeit des Systems sowie die Wiederverwendbarkeit der Fachkomponenten determiniert wird [10]. Einerseits kann die Wiederverwendbarkeit der Fachkomponenten erhöht werden, indem möglichst feingranulare Dienste zusammengefasst werden, die eine sehr spezifische Funktionalität anbieten [6]. Andererseits wird hierdurch jedoch die Anzahl an Fachkomponenten erhöht, was deren Assemblierung in einem Informationssystem erschwert [6].

Um diesen konfliktären Zielen gerecht zu werden, wurde unter anderem die sogenannte Business-Component-Identification-Methode (BCI-Methode) vorgeschlagen, die auf Basis von konzeptionellen Modellen eine heuristische Optimierung der Problemstellung vorsieht [11]. Ein Vorteil der Methode verspricht insbesondere zu sein, dass sie unabhängig von der Art der zugrundeliegenden Modelle verwendet werden kann. Die als Ausgangsbasis verwendeten konzeptionellen Modelle determinieren damit jedoch auch maßgeblich die Qualität der gefundenen Lösung. Dies wirft die zentrale Frage auf, welche Modelle und Notationen sich für die Komponentenfindung besonders gut eignen. Mit Blick auf die Etablierung komplexer, innovativer Geschäftslösungen als soziotechnische Systeme im Rahmen des Business Engineering [12] wächst die Bedeutung einer ganzheitlichen Betrachtung der Modellierung auf der strategischen, konzeptionellen sowie technischen Ebene. Das Semantische Objektmodell (SOM) stellt einen solchen ganzheitlichen Modellierungsansatz dar [13-16]. Die Unternehmensarchitektur des SOM-Ansatzes differenziert die drei Modellebenen Unternehmensplan, Geschäftsprozessmodell und Ressourcenmodell, welche jeweils aus einer struktur- sowie verhaltensorientierten Sichtweise in Form textueller Beschreibungen bzw. semi-formal durch Diagramme beschrieben werden [16]. Während das Vorgehensmodell die Schritte zur Erstellung der einzelnen Artefakte festlegt, stellen Metamodelle die Integration der Sichten sicher. Die konsequente Verfeinerung des Unternehmensplans hin zum Ressourcenmodell in einem Top-Down-Ansatz ermöglicht die Erstellung eines hochwertigen und vor allem hinsichtlich der Anforderungen vollständigen Fachkonzepts, das als Ausgangsbasis wesentlich zum Erfolg der Komponentenfindung beiträgt. Während der SOM-Ansatz also eine etablierte Methodik bereitstellt um das Fachkonzept detailliert zu beschreiben, unterstützt die BCI-Methode die Transformation von Fach- zu Systemkonzept. Trotz dieser vermeintlich symbiotischen Beziehung wurde das Zusammenspiel beider Methoden bisher nicht wissenschaftlich untersucht, weswegen sich insbesondere folgende Forschungsfragen ergeben:

1. Inwiefern eignet sich die SOM-Methodik zur Unterstützung der Modularisierung von Informationssystemen mithilfe der BCI-Methode?
2. Welche SOM-Modelle tragen konkret zur Erlangung valider Ergebnisse bei?

3. Welche Beziehungsformen lassen sich in den SOM-Modellen differenzieren und wie lassen sich diese bei entsprechender Eignung auf die BCI-Methode abbilden?

Zur Beantwortung der Forschungsfragen wird in Abschnitt 2 zunächst der theoretische Grundstock gelegt. Im Rahmen von Abschnitt 3 wird ein Fallbeispiel eingeführt, das die Anwendung der BCI-Methode auf Basis der verschiedenen SOM-Modelle demonstriert. Die Ergebnisse dieser prototypischen Anwendung werden anschließend in Abschnitt 4 diskutiert. Abschließend wird in Abschnitt 5 der Artikel kritisch zusammengefasst und ein Ausblick für weitere Untersuchungen gegeben.

## 2 Theoretische Grundlagen

In den folgenden Teilabschnitten werden grundlegende Begriffe komponentenbasierter Systeme, der BCI-Methode sowie der Modelle des SOM-Ansatzes erläutert.

### 2.1 Komponentenbasierte Systeme

Ein generelles Modell komponentenbasierter Anwendungssysteme wird im Common Business Component Model (COBCOM) erläutert [17]. Neben der Architektur komponentenbasierter Systeme wird hier zudem der Lebenszyklus von Fachkomponenten skizziert. Im Allgemeinen basieren komponentenbasierte Informationssysteme auf einer dreistufigen Client/Server-Architektur. Neben zusätzlichen technischen Anpassungen, die etwa der Vermittlung zwischen Dienstgebern und Dienstnehmern dienen, bspw. mithilfe sogenannter Object Request Broker, ist der zentrale Unterschied die weitere Differenzierung der Verarbeitungslogik [17]. Hierfür wird die betriebliche Funktionalität weiter in sogenannte Fachkomponenten aufgespalten.

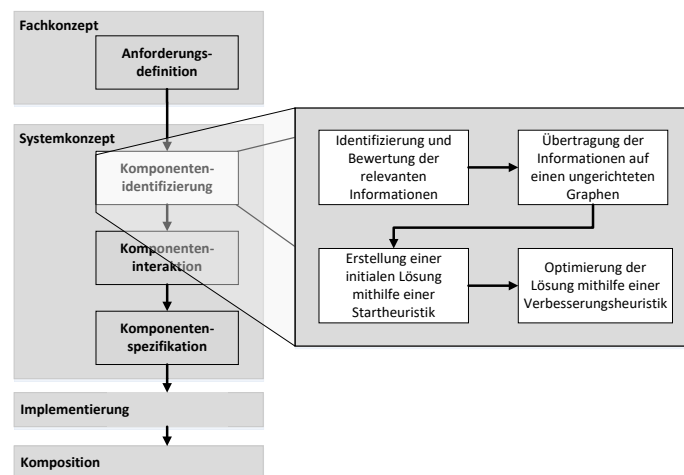


Abbildung 1. Entwicklungsprozess komponentenbasierter Systeme (links) [8] und Ablaufschritte der BCI-Methode (rechts)

**Definition 1 [Fachkomponente]:** „Eine Fachkomponente ist eine Komponente, die eine bestimmte Menge von Diensten einer betrieblichen Anwendungsdomäne anbietet“ [4]. Fraglich ist jedoch, welche Menge an betrieblichen Diensten in einer Fachkomponente zusammengefasst wird. Der Entwicklungsprozess komponentenbasierter Systeme sieht daher auch einen eigenen Schritt zur Identifikation von Fachkomponenten vor (Abbildung 1). Es gilt den Identifikationsprozess so zu gestalten, dass die Anzahl an Schnittstellen zwischen den Fachkomponenten minimiert wird, während die Kohäsion innerhalb der Komponenten maximiert wird [6], [18]. Da die beiden Ziele jedoch konfliktär sind ist die Lösung so zu gestalten, dass beide berücksichtigt werden. Hierfür soll zunächst ein grundsätzliches Verständnis für die beiden Ziele geschaffen werden.

**Definition 2 [Kopplung und Kohäsion]:** Kopplung beschreibt das Ausmaß, in dem die Elemente innerhalb einer Fachkomponente mit Elementen außerhalb zusammenhängen. Kohäsion beschreibt das Ausmaß, in dem die Elemente innerhalb einer Fachkomponente in Wechselbeziehung zueinanderstehen. Die Kohäsion kann dabei als Kopplung innerhalb einer Komponente verstanden und gemessen werden [19], [20].

## 2.2 Methoden zur Identifikation von Fachkomponenten

Bei der Partitionierung eines Informationssystems und der Aufteilung der Dienste auf einzelne Fachkomponenten handelt es sich um ein NP-vollständiges Problem [21]. Zur Lösung dieses Problems wurden in der Literatur diverse Methoden vorgeschlagen und diskutiert [11], [22]. Viele dieser Ansätze schlagen allerdings nur grobe Richtlinien vor [23-27] und sind daher sehr stark abhängig von der Erfahrung des Anwenders. Im Vergleich dazu gibt die BCI-Methode dem Anwender zwar Freiheiten in der Gewichtung der verschiedenen Beziehungen, gibt jedoch auch ein systemgestütztes und einheitliches Vorgehen vor. Weiterhin gibt es diverse Methoden die auf Matrix Analysen basieren [28-30]. Während diese Ansätze zwar leicht anzuwenden sind, können mit ihnen nur zwei Dimensionen untersucht werden, wohingegen die BCI-Methode grundsätzlich beliebig viele Dimensionen berücksichtigt. Abschließend gibt es auch Ansätze, für die kein Tool-Support identifiziert werden konnte, so dass diese Ansätze im Rahmen des Beitrages nicht anzuwenden waren [31], [32]. Folglich wurde auf die BCI-Methode zurückgegriffen. Die Methode geht im Wesentlichen in vier Schritten vor (siehe Abbildung 1): Im ersten Schritt muss der Systemarchitekt entscheiden, welche Informationen aus den konzeptionellen Modellen gewonnen werden können und diese bzgl. ihrer Relevanz bewerten. Typischerweise handelt es sich bei den konzeptionellen Modellen um Geschäftsprozessmodelle, Datenmodelle oder um Funktionsdekompositionsmodelle. In diesem Schritt ist es anschließend notwendig, die verschiedenen Beziehungsarten zu unterscheiden und bezüglich ihrer Relevanz zu bewerten. Hiermit ist dem Systemarchitekten die Möglichkeit gegeben festzulegen, welche Beziehungsarten möglichst innerhalb der Fachkomponenten verlaufen sollen. Es handelt sich dabei um den zentralen Schritt, mit welchem der Systemarchitekt das spätere Ergebnis beeinflussen kann. Im zweiten Schritt werden alle als relevant identifizierten Informationen auf einen ungerichteten Graphen übertragen, wobei alle Elemente als Knoten und alle Beziehungen zwischen ihnen als Kanten abgebildet werden. Im dritten Schritt wird zunächst eine Startlösung generiert. Hierfür wird bspw. mithilfe der „Start-Partition-Greedy“-

Heuristik [11] eine Liste erstellt, welche alle Beziehungen bzgl. ihrer Relevanz sortiert. Am Ende dieses Schrittes steht eine initiale Aufteilung von Elementen zu Fachkomponenten. Im vierten Schritt soll diese Aufteilung optimiert werden, d.h. es soll die Kopplung minimiert und die Kohäsion maximiert werden. Hierfür ist es zunächst notwendig, die beiden Maßzahlen mathematisch zu definieren (Formel 1 und 2):

$$Cou(p) = \frac{1}{2} \sum_{i=1}^k \sum_{u \in C_i} \sum_{v \notin C_i} W_{(u,v)} \quad (1)$$

$$Coh(p) = \frac{1}{2} \sum_{i=1}^k \sum_{u \in C_i} \sum_{v \in C_i, v \neq u} W_{(u,v)} \quad (2)$$

Die Anzahl an Komponenten, die in der initialen Lösung gefunden wurden, wird durch die Variable  $k$  beschrieben.  $w_{(u,v)}$  steht für das Gewicht, welches die Relevanz zwischen den beiden Elementen  $u$  und  $v$  ausdrückt. Im Rahmen des „Kernighan-Lin“-Algorithmus wird anschließend eine Optimierung der gefundenen Startlösung vorgenommen [33]. Einerseits wird die Summe der Gewichte der Beziehungen minimiert, die zwischen verschiedenen Fachkomponenten verlaufen (Formel 3 - links). Andererseits wird die Summe der Gewichte der Beziehungen maximiert, die innerhalb der Fachkomponenten verlaufen (Formel 3 - rechts). Unter der Nebenbedingung, dass die in der Startlösung gefundene Anzahl an Komponenten konstant bleibt, lässt es sich mathematisch nachweisen, dass die beiden Ziele mathematisch äquivalent sind [11].

$$\min_p \frac{1}{2} \sum_{i=1}^k \sum_{u \in C_i} \sum_{v \notin C_i} W_{(u,v)} \Leftrightarrow \max_p \frac{1}{2} \sum_{i=1}^k \sum_{u \in C_i} \sum_{v \in C_i, v \neq u} W_{(u,v)} \quad (3)$$

### 2.3 Modelle des SOM-Ansatzes

Das Semantische Objektmodell (SOM) stellt einen objekt- und geschäftsprozessorientierten Ansatz zur Modellierung betrieblicher Systeme dar und unterscheidet bis auf die Ebene des Unternehmensplans jeweils unterschiedliche Arten von Diagrammen zur Beschreibung von Struktur und Verhalten [15], [16]. Auf Ebene der Geschäftsprozesse bildet die Verknüpfung von betrieblichen Objekten durch Transaktionen im Interaktionsschema (IAS) die Struktur, die Verbindung von Aufgaben (Vorgangstypen) über Ereignisbeziehungen im Vorgangs-Ereignis-Schema (VES) das Verhalten ab. Analog beschreibt das konzeptuelle Objektschema (KOS) auf Ebene des Ressourcenmodells die Struktur des zu spezifizierenden Anwendungssystems durch konzeptuelle Objekttypen (KOT) sowie deren Interaktions-, Generalisierungs- und Aggregationsbeziehungen. Interaktionsbeziehungen zwischen Vorgangsobjekttypen (VOT) spezifizieren schließlich das Verhalten im Vorgangsobjektschema (VOS). Durch Integration der verschiedenen Modelle über die Sichten und Ebenen hinweg lässt sich sicherstellen, dass keine relevanten Informationen bei der Modellierung verloren gehen. Vielmehr findet eine Verfeinerung von einer abstrakten hin zu einer spezielleren Betrachtung auf Anwendungssystemebene statt. Das Vorgehensmodell sichert mit der Vorgabe von Zerlegungsregeln sowie der Integration der Sichten über ein Metamodell explizit die Konsistenz, Vollständigkeit und in der Folge die für die Anwendung der BCI-Methodik entscheidende Qualität des Fachkonzepts.

### 3 Komponentenfindung auf Basis des SOM-Ansatzes

Aufbauend auf den theoretischen Grundlagen wird nachfolgend die BCI-Methode auf Basis eines praxisnahen Fallbeispiels in SOM dargestellt. Der Fokus liegt dabei auf der Identifizierung und Bewertung relevanter Informationen aus den SOM-Modellen.

#### 3.1 Einführung Fallsbeispiel OnlineCars

Das folgende Fallbeispiel stellt vereinfacht die Vermietung und Auftragsabwicklung von Elektrofahrzeugen über ein Internetportal (OnlineCars) dar. Mithilfe der SOM-Methodik wurden schrittweise die Schemata der einzelnen Ebenen erstellt.

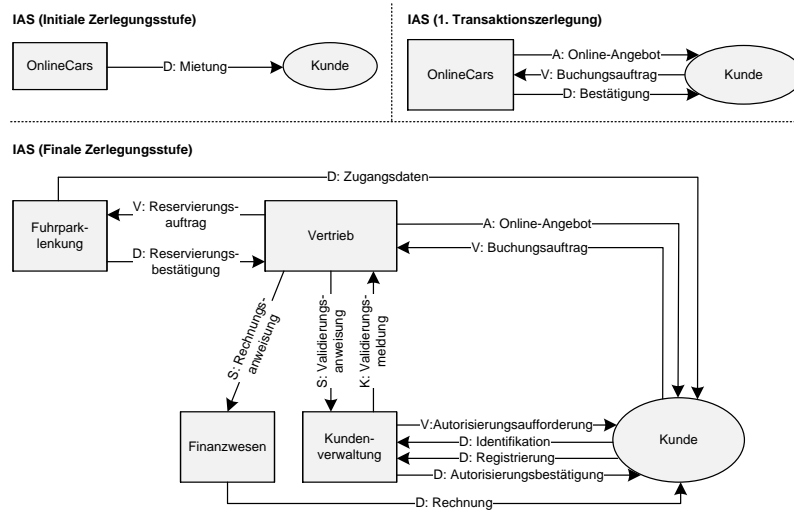
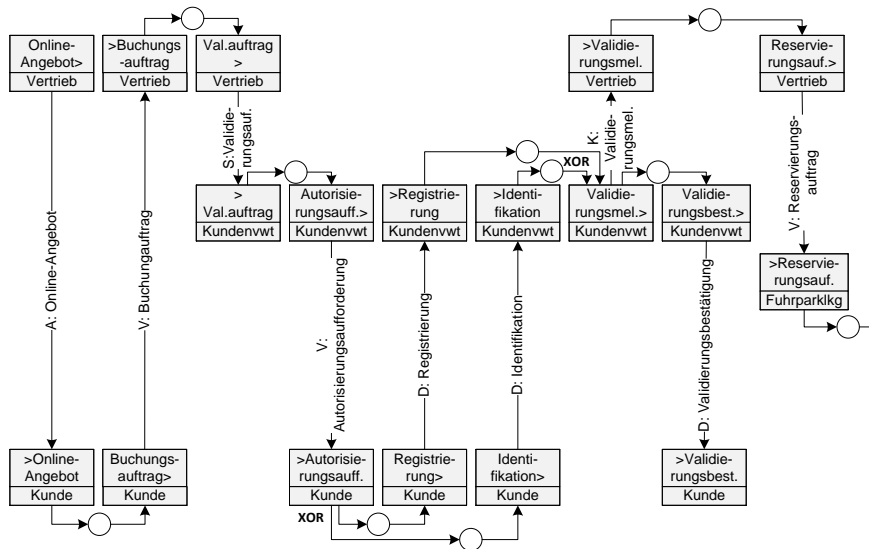


Abbildung 2. IAS des Fallbeispiels

Auf Geschäftsebene wird der Verleihservice in Abbildung 2 strukturorientiert in Form mehrerer Interaktionsschemata (IAS) dargestellt. Ausgehend vom initialen IAS wurden sukzessive Objekt- und Transaktionszerlegungen gemäß den Zerlegungsregeln des SOM-Ansatzes durchgeführt [16]. Während der Vertrieb dem Finanzwesen und der Kundenverwaltung hierarchisch übergeordnet ist, erfolgt die Koordination zwischen Vertrieb und Fuhrparklenkung sowie zwischen Vertrieb und dem Umweltobjekt Kunde auf Basis des Verhandlungsprinzips. Der Verleihservice erstellt als Leistung bei Vorliegen eines Buchungsauftrags neben der Autorisierungsbestätigung und der Rechnung auch die Zugangsdaten und übergibt diese dem Kunden. Die Leistungen werden nach Beauftragung bzw. Anweisung des Vertriebs von unterschiedlichen Objekten erbracht. Der Kunde übermittelt als Leistung aufgrund einer Autorisierungsaufforderung seine Registrierungs- bzw. Identifikationsdaten. In Ergänzung zum IAS beschreibt der Ausschnitt des VES in Abbildung 3 das Verhalten der betrieblichen Objekte im Geschäftsprozess. Der Empfang eines Buchungsauftrags durch den Vertrieb löst bspw. über ein

objektinternes Ereignis die Sendung eines Validierungsauftrags an die Kundenverwaltung aus.



**Abbildung 3.** Ausschnitt VES des Fallbeispiels (finale Zerlegungsstufe)

Die Ableitung der Anwendungssystemspezifikation auf Basis der Geschäftsprozessmodelle IAS/VES in Form des KOS aus Struktursicht und des VOS aus Verhaltenssicht schließt die Modellierung des Anwendungsfalls ab. Abbildung 4 stellt das initiale und das konsolidierte KOS dar. Entsprechend des SOM-Ansatzes werden bei der Konsolidierung diejenigen KOTs entfernt, die mit nicht-automatisierten Aufgaben und Transaktionen korrespondieren [16]. Zudem werden KOTs, die sich in ihren Attributen und/oder Operatoren weitestgehend decken aus Redundanzgründen zusammengefasst. Im Fallbeispiel entstehen dabei die KOTs Autorisierung (*Validierungsauftrag*, *Autorisierungsaufforderung*) und Reservierung (*Reserverierungsauftrag* und *-bestätigung*). Es wurden keine KOTs entfernt, da sämtliche Aufgaben bzw. Transaktionen von dem spezifizierten Anwendungssystem unterstützt werden sollen. Zusätzlich werden im konsolidierten KOS die Beziehungsarten Interaktion (*interacts\_with*), Aggregation (*is\_part\_of*) und Generalisierung (*is\_a*) berücksichtigt.

Das für jedes betriebliche Objekt erstellte VOS wird in einer integrierten Darstellung in Abbildung 5 zusammengeführt. Analog zum KOS werden im VOS diejenigen VOTs entfernt, deren zugehörige Aufgaben nicht automatisiert werden. Ebenso werden VOTs zusammengefasst, deren Aufgaben aus Gründen der semantischen Integrität stets gemeinsam durchzuführen sind. Aus Gründen der Übersichtlichkeit wurden für das KOS und VOS des Fallbeispiels keine weitere Spezifikation, wie bspw. Zuordnung von Operatoren oder Definition von Nachrichten, vorgenommen. Die erzeugten Modelle auf der Geschäftsprozessebene (IAS/VES) sowie auf der Ebene der Anwendungssystemspezifikation (KOS/VOS) bilden die Grundlage für die Identifikation der relevanten Informationen im nächsten Abschnitt.

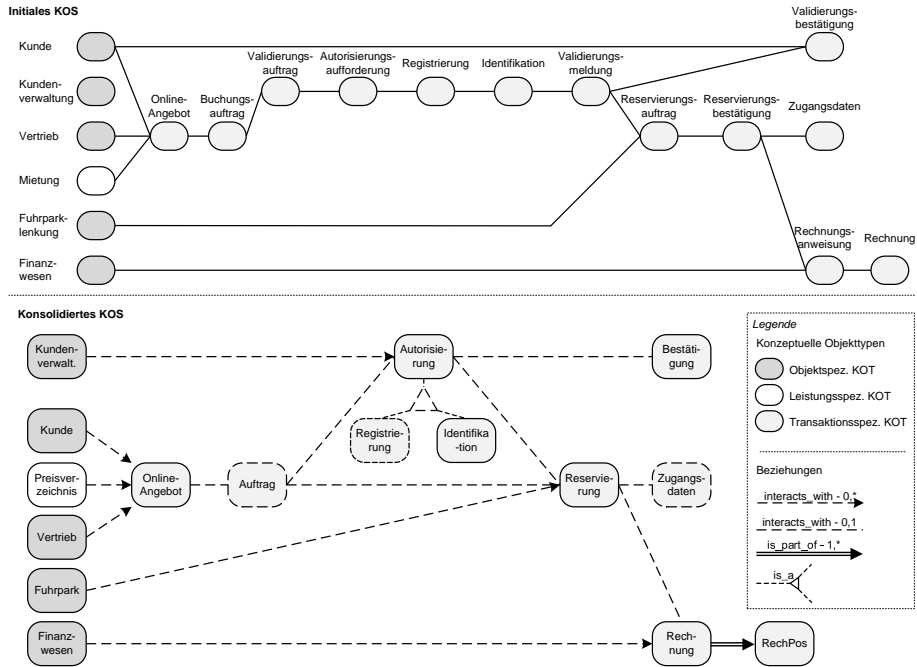


Abbildung 4. Initiales und konsolidiertes KOS des Fallbeispiels

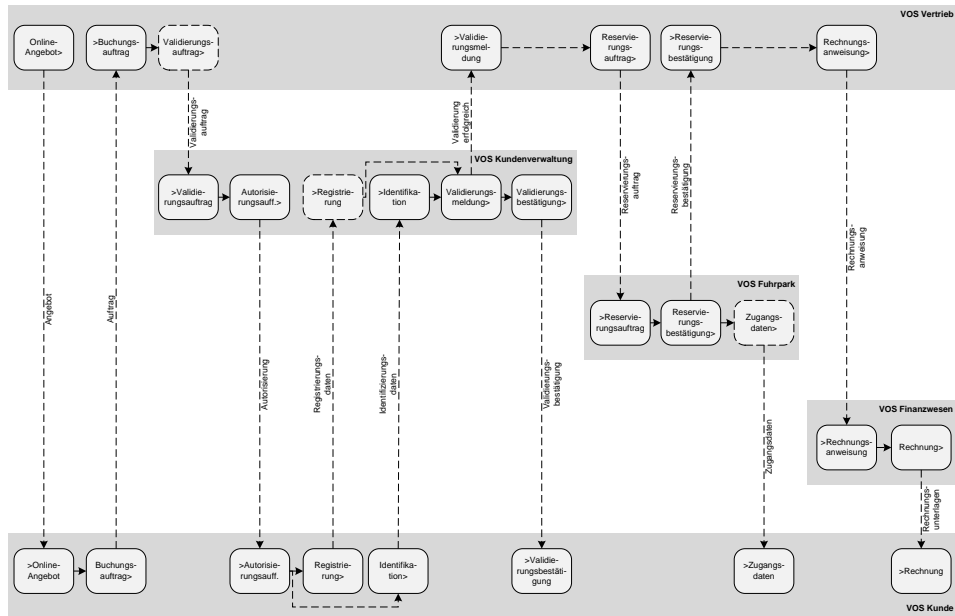


Abbildung 5. Konsolidiertes VOS des Fallbeispiels



### 3.2 Identifizierung und Bewertung der relevanten Informationen

Die Ergebnisse der Modellierung des Anwendungsfalls verdeutlichen bereits die Herausforderung bei der Entscheidung, welche Informationen bei der BCI-Methode zu berücksichtigen sind. Zum einen beziehen sich die Modelle des SOM-Ansatzes auf unterschiedliche Ebenen. Zum anderen wird innerhalb einer Ebene zwischen struktur- und verhaltensorientierten Modellen unterschieden. Zudem bauen die Modelle konsequent aufeinander auf und beschreiben damit ähnliche Sachverhalte.

Im Unterschied zu IAS und VES beinhalten KOS und VOS ausschließlich automatisierbare Elemente, wodurch Komponenten hervorgebracht werden können, die sich auch tatsächlich realisieren lassen. Gleichwohl entsteht aufgrund der Ableitung von KOS und VOS aus IAS und VES kein Informationsverlust, vielmehr wird eine Verfeinerung und Ergänzung hin zur Anwendungsspezifikation vorgenommen [16]. KOS und VOS bilden somit das Fachkonzept als Ausgangsbasis für die Komponentenfindung. In der systematischen Ausrichtung auf das Anwendungssystem als automatisierter Teil des betrieblichen Systems kann somit eine der Stärken von SOM ausgenutzt werden.

In einem weiteren Schritt werden die in KOS/VOS enthaltenen Beziehungen im Hinblick auf ihre Relevanz bewertet. Aufgrund der engen Kopplung von KOTs werden die Aggregations- und Generalisierungsbeziehungen im KOS relevanter eingestuft als Interaktionsbeziehungen zwischen KOTs. Zumindest theoretisch ließen sich zudem objektspezifische, leistungsspezifische und transaktionsspezifische KOTs differenzieren. Allerdings lässt sich nur schwer eine logische Begründung für die Einstufung der Relevanz geben. Interaktionsbeziehungen zwischen KOTs werden nicht berücksichtigt. Die Interaktionsbeziehungen im VOS lassen sich in Anlehnung an das VES nach objektinternen (objektinterne Ereignisse) und objektexternen Beziehungen (Transaktionen) differenzieren. Objektinterne Beziehungen verbinden Aufgaben bzw. VOTs innerhalb eines bestimmten betrieblichen Objekts und werden für die Komponentenfindung daher relevanter angesehen als objektexterne Beziehungen, die Aufgaben bzw. VOTs unterschiedlicher betrieblicher Objekte verbinden. Optionale Beziehungen (XOR), wie sie bspw. zwischen den VOTs *>Autorisierungsaufforderung* und *Registrierung* bzw. *Identifikation* bestehen, wird die geringste Relevanz zugewiesen. Objektexterne Beziehungen können als Teil einer Transaktion einer bestimmten Phase (Anbahnung, Vereinbarung, Durchführung) zugeordnet werden. Die Bewertung der Relevanz der verschiedenen Phasen lässt sich auch hier schwer logisch begründen, so dass objektexterne Beziehungen ohne weitere Differenzierung aufgenommen werden.

### 3.3 Übertragung der identifizierten Informationen auf einen Graphen

Die relevanten Informationen aus KOS und VOS müssen vor der Anwendung von Heuristiken der BCI-Methode auf einen Graphen übertragen werden. KOTs und VOTs werden als Elemente in Form von Knoten im Graphen abgebildet. Die Beziehungen zwischen den KOTs und zwischen den VOTs werden als ungerichtete Kanten dargestellt. Die fehlende Angabe der expliziten Zuordnung zwischen KOTs und VOTs in einem der Modelle führt zunächst zur Entstehung zweier unverbundener Teilgraphen. Es ist daher zur Erzeugung eines zusammenhängenden Graphen zu klären, welche KOTs mit

welchen VOTs in Beziehung stehen. Grundsätzlich ist jeder VOT einem betrieblichen Objekt zugeordnet, dessen Name mit dem Namen des zugehörigen VOS korrespondiert. Die betrieblichen Objekte korrespondieren wiederum mit den entsprechenden objektspezifischen KOTs im KOS. Zunächst werden daher die einzelnen VOTs mit den objektspezifischen KOTs verbunden. Im Fallbeispiel korrespondiert bspw. der VOT *Online-Angebot* mit dem VOS *Vertrieb*. Das VOS *Vertrieb* korrespondiert wiederum mit dem betrieblichen Objekt *Vertrieb* und dem objektspezifischen KOT *Vertrieb*. Es lässt sich damit eine Beziehung zwischen dem VOT *Online-Angebot* und dem KOT *Vertrieb* herstellen. Aufgrund der im KOS durchgeführten Konsolidierung besteht jedoch die Möglichkeit, dass KOTs umbenannt oder mit anderen KOTs zusammengefasst wurden. Auf Grundlage der vorgenommenen Konsolidierungsschritte kann so bspw. der VOT *Rechnungsanweisung* dem KOT *Rechnung*, welcher durch die Zusammenfassung der KOTs *Rechnungsanweisung* und *Rechnung* entstanden ist, zugeordnet werden. Nach der Verbindung der VOTs mit den KOTs veranschaulicht der in Abbildung 6 dargestellte Graph das Ergebnis der Übertragung der Objekte und Beziehungen.

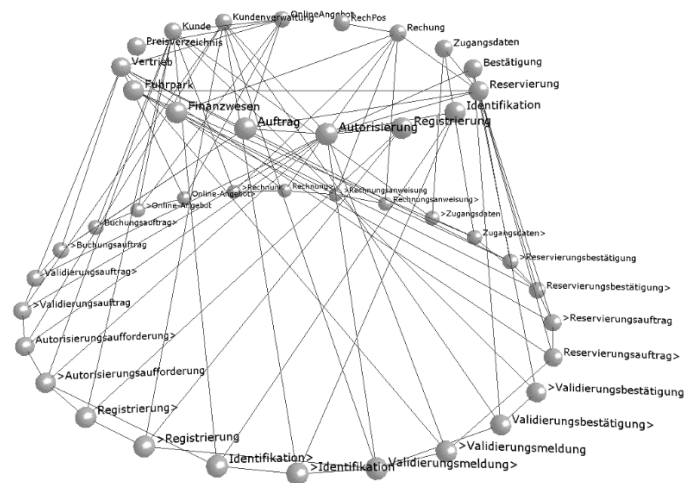
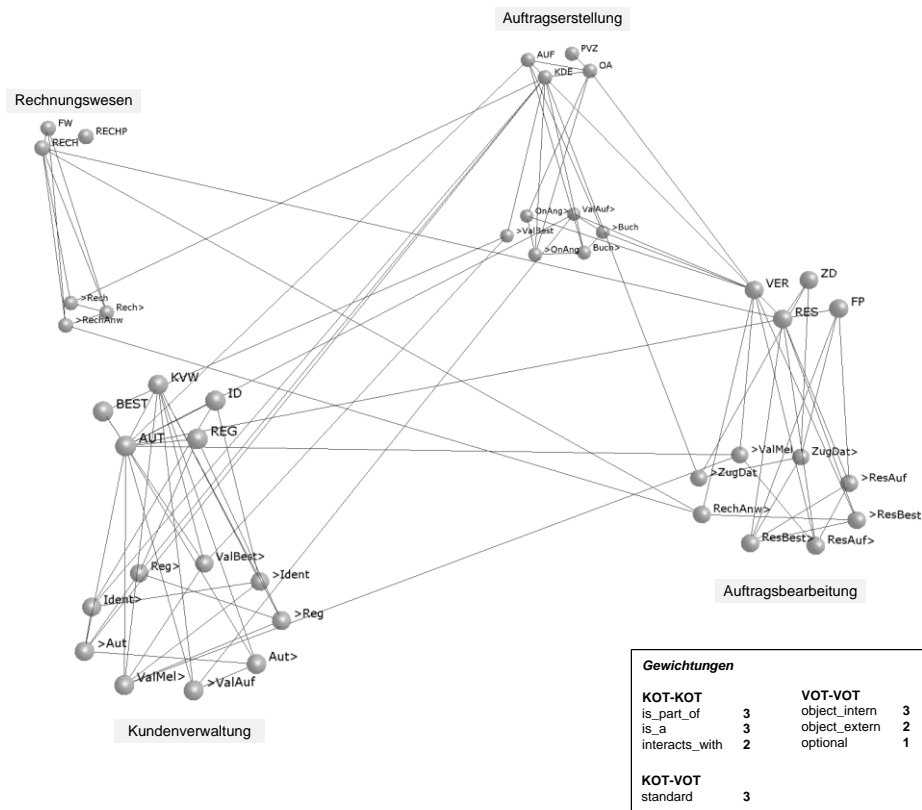


Abbildung 6. Graph der BCI-Methode

## 4 Ergebnisse des Fallbeispiels

Das Ergebnis der Komponentenfindung ist in Abbildung 7 als optimale Partitionierung des Graphen hinsichtlich der angewandten Heuristiken dargestellt. Die Aufteilung der KOTs und VOTs auf die verschiedenen Komponenten wurde durch die Anwendung der Start-Partition-Greedy-Startheuristik in Verbindung mit der Kernighan-Lin-Verbesserungsheuristik erreicht. Grundlage der Anwendung der Heuristiken bildet die Gewichtung der Beziehungen im Graphen. Dabei besteht die Herausforderung darin, die in Abschnitt 3.2 beschriebene unterschiedliche Relevanz der jeweiligen Beziehungen als konkrete numerische Gewichte abzubilden. Analog zur Festlegung der Relevanz der Beziehungen bestehen auch hier Freiheitsgrade für den Systemarchitekten. Um eine

möglichst optimale Lösung zu generieren, wurden verschiedene Gewichtungen unter Beachtung der Relevanz der Beziehungen mithilfe eines Software-Tools (BCI-3D) getestet und die Ergebnisse hinsichtlich ihrer Eignung diskutiert. Die dem Ergebnis zugrundeliegenden Gewichtungen können Abbildung 7 entnommen werden.



**Abbildung 7.** Optimierte Partitionierung des Graphen

Sofern keine Abstufungen aufgrund einer Rangfolge vorgesehen sind, wurde ein Standardgewicht von 3 gewählt. Folglich erhalten bspw. die den objektinternen Beziehungen untergeordneten objektexternen Beziehungen eine geringere Gewichtung von 2. Dem schließen sich die optionalen Beziehungen mit einem Gewicht von 1 an. Insgesamt wurden mit der vorgeschlagenen Gewichtung vier verschiedene Komponenten identifiziert. Jede Komponente fasst im Sinne der Erzielung der maximalen Kohäsion zusammenhängende KOTs und VOTs zusammen und wird in Bezug auf die Gemeinsamkeiten der bereitgestellten Dienste mit einem entsprechenden Namen versehen. Die Komponente *Auftragserstellung* veröffentlicht das Angebot an Fahrzeugen und erstellt die Buchungsaufträge der Kunden. Die Verwaltung der Kunden sowie deren Autorisierung durch Identifikation und Registrierung ist Aufgabe der Komponente *Kundenver-*

waltung. Die Reservierung von Fahrzeugen aus dem Fuhrpark, die Bestätigung der Reservierung sowie die Übermittlung der Zugangsdaten wird von der Komponente *Auftragsbearbeitung* angeboten. Schließlich stellt die Komponente *Rechnungswesen* die Funktionalität für die buchhalterische Erfassung des Auftrags und die Erstellung der Rechnungsdokumente bereit. Die verbleibenden Beziehungen zwischen den Komponenten realisieren in Form von Schnittstellen die Kopplung der Komponenten. Betrachtet man die Kosten der Partitionierung, so konnten diese nach Anwendung der Kernighan-Lin-Heuristik von  $C(P) = 101$  der Startlösung auf  $C(P) = 70$  reduziert werden. Konkret ergibt dies ein Kopplungsverhältnis von  $\overline{Cou}(P) \approx 0.20$  sowie entsprechend ein Kohäsionsverhältnis von  $\overline{Coh}(P) \approx 0.80$ . Im Vergleich zur Startlösung konnte damit im Sinne einer Optimierung der Partitionierung das Kopplungsverhältnis um 0.09 verringert und das Kohäsionsverhältnis um 0.09 erhöht werden. Konkret bedeutet dies, dass die Zahl der Verbindungen zwischen den Komponenten im Vergleich zur Startlösung nochmals reduziert werden konnte.

## 5 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wurde aufgezeigt, wie eine Komponentenfindung nach der BCI-Methode auf Basis des SOM-Ansatzes durchgeführt werden kann. Zunächst wurden hierfür die Grundlagen der Komponentenfindung, die Vorgehensweise der BCI-Methode sowie die Modelle des SOM-Ansatzes erläutert. Anschließend wurde die Anwendung der BCI-Methode auf Basis des SOM-Ansatzes anhand eines praxisnahen Fallbeispiels dargestellt. Die Ergebnisse des Fallbeispiels zeigen, wie ausgehend von den verschiedenen Modellen des SOM-Ansatzes ein Anwendungssystem in eine möglichst optimale Menge an Fachkomponenten unterteilt werden kann. In Bezug auf die in diesem Beitrag adressierten Forschungsfragen lassen sich folgende Rückschlüsse ziehen. Der objekt- und prozessorientierte SOM-Ansatz stellt mit seinen verschiedenen, aufeinander aufbauenden Modellen eine geeignete Ausgangsbasis für die Komponentenfindung dar. Für die Komponentenfindung eignen sich hierbei insbesondere die Modelle auf der Ebene der Anwendungssystemspezifikation, da diese die von einem Anwendungssystem zu unterstützenden Informationen beinhalten. Im Hinblick auf die Unterscheidung und Übertragung der für die Komponentenfindung relevanten Beziehungen bietet der SOM-Ansatz verschiedene Möglichkeiten. Für das Fallbeispiel wurden bspw. die verschiedenen Beziehungsarten innerhalb des KOS als unterschiedlich relevant für die Komponentenfindung angesehen. Ebenso wurden die Interaktionsbeziehungen innerhalb des VOS weiter in objektinterne, objektexterne und optionale Beziehungen differenziert und in dieser Reihenfolge mit einer absteigenden Relevanz für die Komponentenfindung versehen. Durch die Trennung von strukturorientierten und verhaltensorientierten Modellsichten innerhalb des SOM-Ansatzes ergibt sich bei der Übertragung der relevanten Informationen auf einen Graphen die Problematik, dass hierbei zwei nicht miteinander verbundene Teilgraphen entstehen. Um dieser Problematik entgegenzuwirken, wurde innerhalb des Beitrags eine Möglichkeit aufgezeigt, wie beide Teilgraphen auf Grundlage der Informationen aus KOS und VOS zu einem gemeinsamen Graphen zusammengeführt werden können. Durch das Erstellen einer

initialen Lösung mittels einer Startheuristik und der Optimierung dieser Lösung mittels einer Verbesserungsheuristik konnte eine möglichst optimale Aufteilung von vier Fachkomponenten ermittelt werden. Jede Fachkomponente wird dabei von einer Menge aus KOTs und VOTs repräsentiert, die gegenüber den KOTs und VOTs der anderen Fachkomponenten eine maximale Kohäsion und eine minimale Kopplung aufweisen.

Aus den Ergebnissen und Limitationen des Beitrags ergeben sich sowohl für die Forschung als auch für die Praxis mehrere Implikationen. Innerhalb des Beitrags wurde ein einzelnes, vereinfachtes Fallbeispiel betrachtet, bei welchem aus Gründen der Einfachheit und Übersichtlichkeit keine weiterführende Spezifikation der SOM-Modelle auf der Ebene der Anwendungsspezifikation vorgenommen wurde. Aus Sicht der Praxis bezieht sich der Nutzen vor allem auf die Möglichkeit, systematisch Komponenten auf Grundlage eines objektorientierten Ansatzes wie SOM identifizieren zu können. Mithilfe des im Beitrag vorgestellten Vorgehens wird erstmals eine methodische Grobstruktur bereitgestellt, die erläutert in welchen Schritten SOM-Modelle zur Komponentenfindung mit der BCI-Methode eingesetzt werden können. Weiterhin kann in zukünftigen praxisnahen Forschungsarbeiten die Unterscheidung der Beziehungen sowie die Gewichtungen der Beziehungsarten diskutiert werden, um eine konsolidierte Unterscheidung der Beziehungen zu erreichen. Auf Basis eines solchen konsolidierten Ergebnisses kann ein Systemarchitekt weitaus leichter Anpassungen entsprechend seiner persönlichen Präferenzen vornehmen oder auch die Sensitivität der Ergebnisse prüfen. Zudem kann mithilfe des verwendeten Tools, das die BCI-Methode umsetzt, der Ansatz auch in größeren und industrienahen Projekten verwendet werden. Insbesondere bei solchen Projekten verspricht die vorgestellte Methode mithilfe des Tools eine Vielzahl an Vorteilen gegenüber einer manuellen Systemaufteilung mit sich zu bringen.

Aus Sicht der Forschung könnte bspw. die Übertragung des Vorgehens auf weitere, komplexere Fallbeispiele einen Mehrwert dahingehend bieten, die Eignung der im Beitrag ausgewählten Beziehungen inkl. deren Relevanz zu überprüfen. In diesem Zusammenhang könnte es ebenfalls von Interesse sein, weitere Differenzierungen der Beziehungsarten zu diskutieren und diese hinsichtlich ihrer Praktikabilität zu untersuchen. Potenzial besteht im Besonderen mit Blick auf die Zusammenführung von KOS und VOS bei der Berücksichtigung der Zuordnung von Operatoren zu einzelnen VOTs sowie bei der Definition von Nachrichtenbeziehungen als Erweiterung der Anwendungsspezifikation in SOM. Grundsätzlich ist die Integration der Modellierungssichten über ein Metamodell eine der Stärken des SOM-Ansatzes. Im Zusammenspiel mit der Vorgabe definierter Zerlegungsregeln in Form eines Vorgehensmodells kann die Konsistenz sowie die Vollständigkeit des Fachkonzepts sichergestellt werden. Die systematische Ausrichtung auf das Anwendungssystem als automatisierter Teil ist weiterhin kongruent zu den Zielen der BCI-Methode. Inwiefern Anwender hierdurch zu hochwertigeren Fachkonzepten gelangen, und damit im Rahmen der BCI-Methode bessere Ergebnissen bzgl. Kohäsion und Kopplung erzielen, müsste jedoch in zukünftigen Forschungsarbeiten empirisch geprüft werden. Weiterhin können als Resultat solcher Diskussionen SOM-Modelle auch in anderen Komponentenfindungsverfahren verwendet werden, die ebenfalls zwischen den diskutierten Beziehungen unterscheiden müssen.

Ein Merkmal zur Reduktion der Abhängigkeiten bei der Komponentenorientierung bildet die Erhöhung des Abstraktionsniveaus bei der Entwicklung, da Komponenten in

Bezug zum Klassenkonzept der Objektorientierung mehr Funktionalität kapseln und dadurch abstraktere Entwurfseinheiten darstellen [34], [35]. Hinsichtlich der Gemeinsamkeit zwischen Klassen und Komponenten durch die Kapselung von Funktionalität konnte im vorliegenden Beitrag aufgezeigt werden, wie sich systematisch mithilfe der BCI-Methode Elemente des objektorientierten SOM-Ansatzes zu semantisch sinnvollen, abstrakteren Einheiten in Form von Komponenten zusammenfassen lassen. Die Ergebnisse des SOM-Ansatzes stellen dabei mit der bereits vorhandenen Abgrenzung einzelner Objektklassen eine sehr gute Ausgangsbasis für den Ausbau der Anwendungsspezifikation hin zu einem komponentenbasierten System dar. Während die SOM-Modelle einen bewährten Ansatz liefern, um das Fachkonzept detailliert zu beschreiben, kann darauf aufbauend die BCI-Methode verwendet werden, um das Systemkonzept abzuleiten. Die BCI-Methode liefert eine systematische Unterstützung für den Übergang vom Fach- zum Systemkonzept durch die Identifikation geeigneter Fachkomponenten, ist hierbei jedoch auf hochwertige Modelle im Fachkonzept angewiesen. Insofern konnte im vorliegenden Beitrag gezeigt werden, dass sich die beiden Ansätze ideal ergänzen und weiteren Boden für zukünftige Forschungsarbeiten liefern.

## Literatur

1. McDonald, M., Begin, J., Fortino, S.: Meeting the challenge: the 2009 CIO agenda. Gartner, Inc. (2009)
2. Herzum, P., Sims, O.: Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise. John Wiley & Sons, NY (2000)
3. Vitharana, P., Zahedi, F., Jain, H.: Knowledge-Based Repository Scheme for Storing and Retrieving Business Components: A Theoretical Design and an Empirical Analysis. IEEE Transactions on Software Engineering 29, 649-664 (2003)
4. Turowski, K.: Fachkomponenten: Komponentenbasierte betriebliche Anwendungssysteme. Shaker, Aachen (2003)
5. Sharp, J., Ryan, S.: A Review of Component-Based Software Development. In: 26th International Conference on Information Systems (ICIS) (2005)
6. Szyperski, C., Gruntz, D., Murer, S.: Component Software: Beyond Object-Oriented Programming. Addison-Wesley, Harlow (2002)
7. Brown, A.W.: Large-Scale, Component-Based Development. Prentice Hall, Upper Saddle River, NJ (2000)
8. Cheesman, J., Daniels, J.: UML Components: A Simple Process for Specifying Component-Based Software. Addison-Wesley, Upper Saddle River, NJ (2001)
9. Lim, W.C.: Managing Software Reuse - A Comprehensive Guide to Strategically Reengineering the Organization for Reusable Components. Prentice Hall, Upper Saddle River, NJ (1998)
10. Gorton, I.: Essential Software Architecture. Springer, Heidelberg (2011)
11. Birkmeier, D., Overhage, S.: A Method to Support a Reflective Derivation of Business Components from Conceptual Models. ISeB 11, 403-435 (2012)
12. Österle, H., Blessing, D.: Ansätze des Business Engineering. HMD - Praxis der Wirtschaftsinformatik 241, 7-17 (2005)
13. Ferstl, O.K., Sinz, E.J.: Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell. WIRTSCHAFTSINFORMATIK 32, 566-581 (1990)

14. Ferstl, O.K., Sinz, E.J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM). BISE 33, 477-491 (1991)
15. Ferstl, O.K., Sinz, E.J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. BISE 37, 209-220 (1995)
16. Ferstl, O.K., Sinz, E.J.: Grundlagen der Wirtschaftsinformatik. Oldenbourg, München (2013)
17. Rautenstrauch, C., Turowski, K.: Common Business Component Model (COBCOM): Generelles Modell komponentenbasierter Anwendungssysteme. In: 5. Internationale Tagung Wirtschaftsinformatik, pp. 681-695. (2001)
18. Parnas, D.L.: On the Criteria to be Used in Decomposing Systems into Modules. Communications of the ACM 15, 1053-1058 (1972)
19. Vitharana, P., Jain, H., Zahedi, F.: Strategy-Based Design of Reusable Business Components. IEEE Trans Syst Man Cybern 34, 460-474 (2004)
20. Wallnau, K.C., Hissam, S.A., Seacord, R.C.: Building Systems from Commercial Components. Addison-Wesley, Upper Saddle River, NJ (2002)
21. Garey, M., Johnson, D., Stockmeyer, L.: Some Simplified NP-Complete Problems. In: 6th Annual ACM Symposium on Theory of Computing, pp. 47-63 (1974)
22. Birkmeier, D., Overhage, S.: On Component Identification Approaches – Classification, State of the Art, and Comparison. In: Lewis, G.A., Poernomo, I., Hofmeister, C. (eds.) Component-Based Software Engineering, vol. 5582, pp. 1-18. Springer, Berlin, Germany (2009)
23. Sugumaran, V., Tanniru, M., Storey, V.: Identifying software components from process requirements using domain model and object libraries. In: 20th International Conference on Information Systems (1999)
24. Levi, K., Arsanjani, A.: A goal-driven approach to enterprise component identification and specification. Communications of the ACM 45, 45-52 (2002)
25. Cui, J., Chae, H.: Applying agglomerative hierarchical clustering algorithms to component identification for legacy systems. Information and Software Technology 43, 601-614 (2011)
26. Kang, K., Kim, S., Lee, J., Kim, K., Shin, E., Huh, M.: FORM: a feature-oriented reuse method with domain-specific reference architectures. Ann Softw Eng 5, 143-168 (1998)
27. Birkmeier, D.Q., Gehlert, A., Overhage, S., Schlauderer, S.: Alignment of Business and IT Architectures in the German Federal Government: A Systematic Method to Identify Services from Business Processes. In: Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS), pp. 3848-3857 (2013)
28. Kim, S., Chang, S.: A systematic method to identify software components. In: 11th Asia-Pacific software engineering conference (2004)
29. Lee, S., Yang, Y., Cho, E., Kim, S., Rhew, S.: COMO: A UML-based component development methodology. In: 6th Asia-Pacific software engineering conference (1999)
30. Jang, Y., Kim, E., Lee, K.: Object-oriented component identification method using the affinity analysis technique. In: 9th Conference OOIS (2003)
31. Ganesan, R., Sengupta, S.: A technique for the design of component-based applications. In: 39th Conference TOOLS (2001)
32. Meng, F., Zhan, D., Xu, X.: Business component identification of enterprise information system: a hierarchical clustering method. In: IEEE international conference on e-Business engineering (2005)
33. Kernighan, B., Lin, S.: An Efficient Heuristic Procedure for Partitioning Graphs. Bell System Technical Journal 49, 291-307 (1970)
34. Sommerville, I.: Software Engineering. Pearson Education, Harlow (2007)
35. Zwintzsch, O.: Software-Komponenten im Überblick: Einführung, Klassifizierung & Vergleich von JavaBeans, EJB, COM+, .Net, CORBA, UML 2. W3L, Herdecke (2005)