

1 Jenseits objektorientierter Entwicklung: Systematische Identifikation von Fachkomponenten mit SOM

Thomas Friedrich, Martin Robel, Sebastian Schlauderer

1.1 Motivation

Die Entwicklung von Informationssystemen steht heutzutage weiterhin vor einer Vielzahl teils konterkarierender Herausforderungen. Einerseits ist es notwendig, auf sich ändernde Marktbedingungen schnell reagieren zu können und flexibel Funktionalität anpassen oder auch hinzufügen zu können (McDonald et al. 2009). Andererseits wächst der Kostendruck auf die Entwicklungsabteilungen (McDonald et al. 2009). Vor diesem Hintergrund verspricht die modulare Entwicklung von Informationssystemen eine Vielzahl an Vorteilen mit sich zu bringen (Herzum und Sims 2000; Vitharana et al. 2003). 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 (Turowski 2003), können weiterhin Entwicklungszeiten und Entwicklungskosten drastisch gekürzt werden (Sharp und Ryan 2005). Im Vergleich zur reinen Objektorientierung kann durch das Zusammenfassen mehrerer Objekte zu einer Fachkomponente das Problem fragiler Basisklassen eingegrenzt werden und gleichermaßen sichergestellt werden, dass durch Vererbungsstrukturen das Geheimnisprinzip nicht durchbrochen wird (Szyperski et al. 2002).

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 (Brown 2000; Cheesman und Daniels 2001; Herzum und Sims 2000; Lim 1998; Szyperski et al. 2002). 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 (Gorton 2011). Einerseits kann die Wiederverwendbarkeit der Fachkomponenten erhöht werden, indem möglichst feingranulare Dienste zusammengefasst werden, die eine sehr spezifische Funktionalität anbieten (Szyperski et al. 2002). Andererseits wird hierdurch jedoch die Anzahl an Fachkomponenten erhöht, was deren Assemblierung in einem Informationssystem erschwert und zu einer deutlich erhöhten Anzahl an Schnittstellen führt, die verwaltet werden müssen (Szyperski et al. 2002).

Um diesen konfliktären Zielen gerecht zu werden, wurde die sogenannte Business-Component-Identification-Methode (BCI-Methode) vorgeschlagen, die auf Basis von konzeptionellen Modellen eine heuristische Optimierung der Problemstellung vorsieht (Birkmeier und Overhage 2012). Grundsätzlich soll die vorgeschlagene Methode unabhängig von der Art der zugrundeliegenden Modelle verwendet werden können. Im Rahmen dieses Artikels wird die Methode daher erstmals auf Basis des Semantischen Objektmodells (SOM) eingesetzt. Die im Rahmen des Artikels adressierten Forschungsfragen lauten wie folgt:

- (i) Inwiefern eignet sich der SOM-Ansatz, um mithilfe der BCI-Methode Informationssysteme zu modularisieren?
- (ii) Welche SOM-Modelle werden im Rahmen der BCI-Komponentenfindung benötigt, um zu einem validen Ergebnis zu gelangen?
- (iii) Welche Beziehungen gilt es in den SOM-Modellen zu unterscheiden und wie werden diese Beziehungen auf die BCI-Methode übertragen?

Zur Beantwortung der Forschungsfragen wird in Abschnitt 2 zunächst der theoretische Grundstock gelegt. Hierfür wird im ersten Schritt ein Ordnungsrahmen für die Entwicklung komponentenbasierter Systeme vorgestellt, bevor die BCI-Methode im Detail erörtert wird. Zudem werden die verschiedenen Modelle des SOM-Ansatzes erläutert. Im Rahmen von Abschnitt 3 wird ein Fallbeispiel eingeführt. Im Zuge des Fallbeispiels wird 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.

1.2 Theoretische Grundlagen

In den folgenden Teilabschnitten werden grundlegende Begriffe und Theorien in Bezug auf die Entwicklung komponentenbasierter Systeme, die BCI-Methode sowie die Modelle des SOM-Ansatzes erläutert.

1.2.1 Komponentenbasierte Systeme

Ein generelles Modell komponentenbasierter Anwendungssysteme wird im Common Business Component Model (COBCOM) erläutert (Rautenstrauch und Turowski 2001). Neben der Architektur komponentenbasierter Systeme wird hier zudem der Lebenszyklus von Fachkomponenten skizziert. Im Allgemeinen basieren komponentenbasierte Informationssysteme auf einer dreistufigen Client/Server-Architektur, welche zwischen den Bestandteilen Verarbeitung, Präsentation und Datenhaltung unterscheidet. Neben zusätzlichen technischen Anpassungen, die etwa der Vermittlung zwischen Dienstgebern und Dienstnehmern, bspw. mithilfe sogenannter Object Request Broker, oder einer Automatisierung der Ablauflogik dienen, ist der zentrale Unterschied die weitere Differenzierung der Verarbeitungslogik (Rautenstrauch und Turowski 2001). Hierfür wird die im Bereich der Verarbeitungslogik angeordnete betriebliche Funktionalität in sogenannte Fachkomponenten aufgespalten.

Definition 1 [Fachkomponente]: „Eine Fachkomponente ist eine Komponente, die eine bestimmte Menge von Diensten einer betrieblichen Anwendungsdomäne anbietet“ (Turowski 2003).

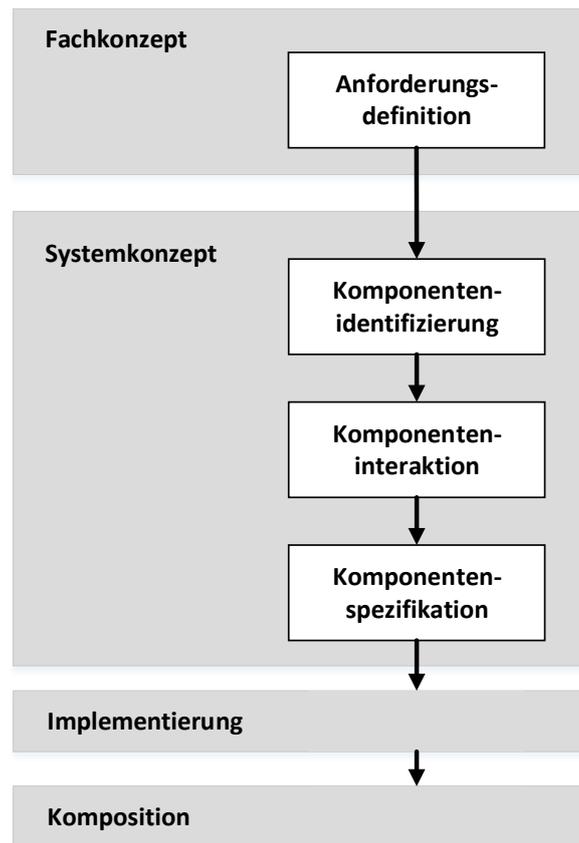


Abb. 1.1 Entwicklungsprozess komponentenbasierter Systeme (in Anlehnung an: Cheesman und Daniels 2001)

Fraglich ist indes, welche Menge an betrieblichen Diensten der betrieblichen Anwendungsdomäne in einer Fachkomponente zusammengefasst werden soll. Der Entwicklungsprozess komponentenbasierter Systeme sieht daher auch einen eigenen Schritt zur Identifikation von Fachkomponenten vor (siehe Abb. 1.1). Hierbei gilt es den Identifikationsprozess so zu gestalten, dass die Anzahl an Schnittstellen zwischen den Fachkomponenten minimiert wird, während die Kohäsion innerhalb der Komponenten maximiert werden soll (Parnas 1972; Szyperski et al. 2002). Da es sich dabei jedoch um konfliktäre Ziele handelt, ist eine Lösung zu finden, die beide Ziele gleichermaßen berücksichtigt. 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 der Komponente 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 (Vitharana et al. 2004; Wallnau et al. 2002).

1.2.2 Methode zur Identifikation von Fachkomponenten

Da bei der Partitionierung eines Informationssystems und der Aufteilung der Dienste auf einzelne Fachkomponenten eine enorme Menge an Lösungen existiert, handelt es sich bei der Lösungsfindung um ein NP-vollständiges Problem (Garey et al. 1974). Um die beiden oben genannten Ziele gleichermaßen zu berücksichtigen, wurde die BCI-Methode entwickelt, welche mithilfe von Heuristiken eine möglichst optimale Lösung sucht (Birkmeier und Overhage 2012). Die Methode geht im Wesentlichen in vier Schritten vor (siehe Abb. 1.2):

- 1) Identifizierung und Bewertung der relevanten Informationen

- 2) Übertragung der identifizierten Informationen auf einen ungerichteten Graphen
- 3) Erstellung einer initialen Lösung mithilfe einer Startheuristik
- 4) Optimierung der Lösung mithilfe einer Verbesserungsheuristik

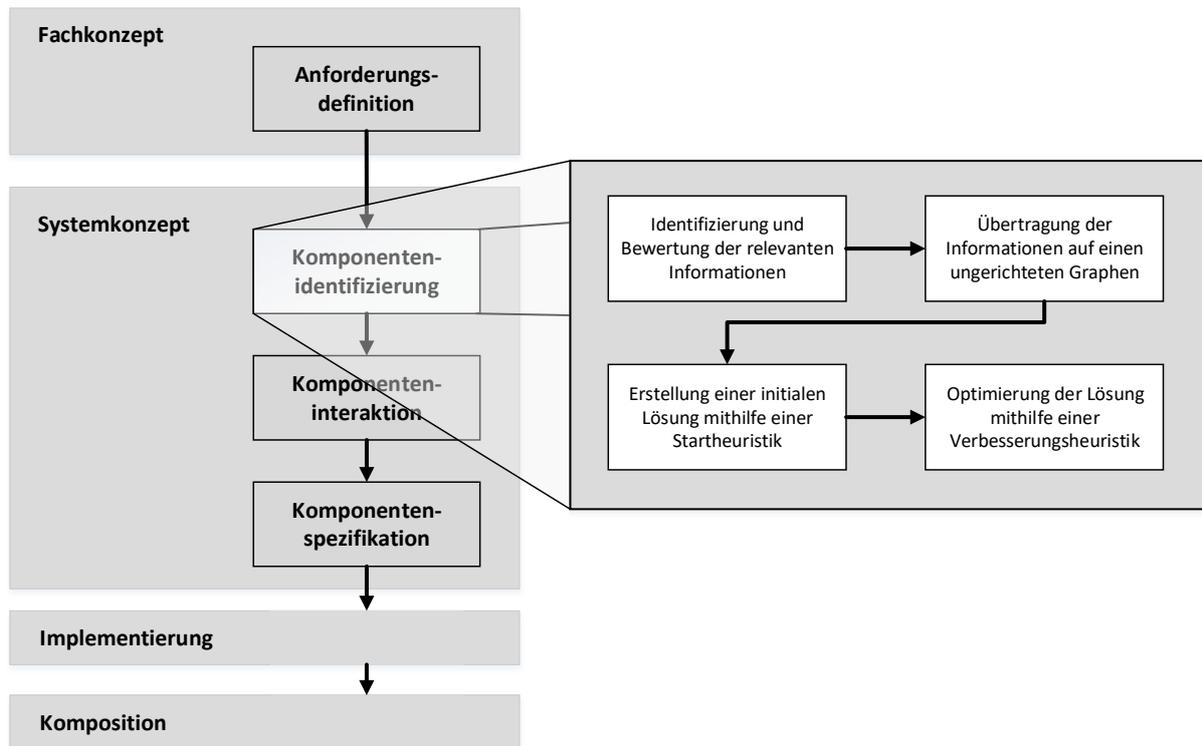


Abb. 1.2 Ablaufschritte der BCI-Methode

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. Hierbei handelt es sich um den zentralen Schritt mit welchem der Systemarchitekt das spätere Ergebnis massiv 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 (Birkmeier und Overhage 2012) eine Liste erstellt, welche alle Beziehungen bzgl. ihrer Relevanz sortiert. Damit kann sichergestellt werden, dass die Beziehungen, die vom Systemarchitekten als besonders relevant empfunden werden, innerhalb einer Fachkomponente verlaufen. 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 (Coupling) minimiert und die Kohäsion (Cohesion) maximiert werden. Hierfür ist es zunächst nötig, die beiden Maßzahlen mathematisch zu definieren (Formel 1 und 2):

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

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

Hierbei steht k für die Anzahl an Komponenten, die in der initialen Lösung gefunden wurden und $w_{(u,v)}$ 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 (Kernighan und Lin 1970). Hierfür wird einerseits 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). Da die Summe der Gewichte aller Beziehungen insgesamt immer konstant bleibt, lässt es sich mathematisch nachweisen, dass die beiden Ziele mathematisch äquivalent sind (Birkmeier und Overhage 2012).

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

1.2.3 Modelle des SOM-Ansatzes

Das Semantische Objektmodell (SOM) ist ein objekt- und geschäftsprozessorientierter Ansatz zur Modellierung betrieblicher Systeme (Ferstl und Sinz 2013). Aus Gründen der Handhabbarkeit und zur Unterstützung der Komplexitätsbewältigung werden betriebliche Systeme mit dem SOM-Ansatz aus verschiedenen Blickwinkeln modelliert. In diesem Abschnitt wird kurz auf die unterschiedlichen Modelle des SOM-Ansatzes eingegangen. Zudem wird eine erste Einschätzung der Modelle hinsichtlich deren Eignung für die Komponentenfindung vorgenommen. Die folgenden Erläuterungen zu den Modellen des SOM-Ansatzes beziehen sich auf Ferstl und Sinz (1990), Ferstl und Sinz (1991), Ferstl und Sinz (1995) sowie Ferstl und Sinz (2013). Für eine ausführliche Beschreibung des SOM-Ansatzes sei auf Kapitel 8 verwiesen.

Die Unternehmensarchitektur des SOM-Ansatzes differenziert ausgehend von einer Top-Down-Betrachtung zwischen drei Modellebenen: (1) Unternehmensplan, (2) Geschäftsprozessmodell sowie (3) Ressourcenmodell. Entsprechend dem Vorgehensmodell des SOM-Ansatzes wird jede dieser Modellebenen in Form einer strukturorientierten und einer verhaltensorientierten Modellsicht spezifiziert. Die Struktursicht des Unternehmensplans wird durch das Objektsystem, die Verhaltenssicht durch das Zielsystem abgebildet. Das Geschäftsprozessmodell wird aus der Struktursicht in Form eines Interaktionsschemas (IAS) und aus der Verhaltenssicht in Form eines Vorgangs-Ereignis-Schemas (VES) modelliert. Die Spezifikation des Ressourcenmodells, welche sich in diesem Zusammenhang auf die Spezifikation von Anwendungssystemen bezieht, wird aus der Struktursicht in Form eines Konzeptuellen Objektschemas (KOS) und aus der Verhaltenssicht in Form eines Vorgangsobjektschemas (VOS) beschrieben.

Für die Modellierung von Objekt- und Zielsystem werden in der Regel informale Darstellungsformen (textuelle Beschreibungen) verwendet. Die Modellierung der Geschäftsprozesse (IAS/VES) sowie die Spezifikation der Anwendungssysteme (KOS/VOS) erfolgt über semi-formale Darstellungsformen (Diagramme). Die methodische Grundlage für die Modellierung der Geschäftsprozesse sowie die Spezifikation der Anwendungssysteme bildet jeweils ein integriertes Metamodell, anhand dessen die beiden struktur- und verhaltensorientierten Sichten der jeweiligen Modellebene abgeleitet werden. Das Metamodell legt hierbei die für die Modellierung verfügbaren Bausteine und deren Beziehungen fest. Tab. 1.1 gibt zusammenfassend einen Überblick über die unterschiedlichen Modelle des SOM-Ansatzes und beschreibt deren typische Inhalte.

Tab. 1.1 Modelle des SOM-Ansatzes und deren Inhalte

<i>Ebene</i>	<i>Sicht</i>	<i>Modell</i>	<i>Inhalte</i>
(1)	Struktur	Objektsystem	Beschreibung der relevanten Diskurswelt und Umwelt sowie zugehöriger Leistungsbeziehungen.
	Verhalten	Zielsystem	Beschreibung von Sach- und Formalzielen sowie von Strategien und Rahmenbedingungen für deren Umsetzung.
(2)	Struktur	Interaktions-schemata (IAS)	Spezifikation der Struktur von Geschäftsprozessen in Form von betrieblichen Objekten, die durch Transaktionen verknüpft sind.
	Verhalten	Vorgangs-Ereignis-Schema (VES)	Spezifikation des Verhaltens von Geschäftsprozessen in Form von Aufgaben (bzw. Vorgangstypen) und ihren Ereignisbeziehungen.
(3)	Struktur	Konzeptuelles Objektschema (KOS)	Spezifikation der Struktur (z.T. auch Verhalten) von Anwendungssystemen in Form von konzeptuellen Objekttypen (KOT) und ihren Interaktions-, Generalisierungs- bzw. Aggregationsbeziehungen.
	Verhalten	Vorgangsobjekt-schemata (VOS)	Spezifikation des Verhaltens von Anwendungssystemen in Form von Vorgangsobjekttypen (VOT) und ihren Interaktionsbeziehungen.

Wie in Abschnitt 1.2.2 beschrieben, werden die für die Komponentenfindung benötigten Informationen typischerweise aus konzeptionellen Modellen gewonnen. Zu den in die Komponentenfindung einbezogenen Informationen zählen bspw. Prozessschritte, Daten und Funktionen. Ebenso bilden Informationen über die Beziehungen zwischen diesen Objekten eine wesentliche Grundlage. Im Hinblick auf den SOM-Ansatz ist demnach zu untersuchen, aus welchen Modellen welche Informationen für die Komponentenfindung gewonnen werden können.

Wenngleich die Modelle der ersten Ebene wichtige Informationen über das betriebliche System aus einer globalen Perspektive liefern, sind diese Informationen für die Komponentenfindung noch zu abstrakt und zu unspezifisch. Beispielsweise können aus diesen Modellen keine Informationen über die verschiedenen Beziehungsarten gewonnen werden. Ebenso erschwert die textuelle Darstellungsweise die Identifikation relevanter Objekte. Die Modelle der ersten Ebene werden für die Komponentenfindung daher als weniger relevant eingestuft. Aufgrund ihrer spezifischeren Informationen sowie ihrer semi-formalen Darstellungsweise können sowohl die Modelle der zweiten Ebene als auch die Modelle der dritten Ebene für die Komponentenfindung als geeignet angesehen werden. In Bezug auf die Modelle der zweiten Ebene lassen sich bspw. Informationen zur Struktur von Geschäftsprozessen in Form von betrieblichen Objekten und deren Transaktionsbeziehungen aus dem IAS entnehmen. Analog können verhaltensrelevante Informationen in Form von Aufgaben und deren Ereignisbeziehungen aus dem VES entnommen werden. Zu berücksichtigen gilt jedoch, dass die in diesen Modellen enthaltenen Informationen sich auch auf nicht-automatisierbare Aufgaben bzw. Transaktionen beziehen. Eine Abgrenzung hinsichtlich der von einem Anwendungssystem zu unterstützenden Elemente wird in den Modellen der dritten Ebene vorgenommen. Innerhalb des KOS, welches als objektorientierte Erweiterung und Modifikation eines konzeptuellen Datenschemas angesehen werden kann, werden die einem Anwendungssystem zugeordneten betrieblichen Objekte in Form von konzeptuellen Objekttypen (KOT) und deren Beziehungen spezifiziert. Innerhalb des VOS wiederum wird das Zusammenwirken von KOTs in Bezug auf die Durchführung einer betrieblichen Aufgabe in Form von Vorgangsobjekttypen (VOT) und deren Beziehungen beschrieben. KOS und VOS bilden zusammen die fachliche Spezifikation eines Anwendungssystems.

1.3 Komponentenfindung auf Basis des SOM-Ansatzes

Aufbauend auf den theoretischen Grundlagen wird nachfolgend die Anwendung der Komponentenfindung auf Basis des SOM-Ansatzes anhand eines praxisnahen Fallbeispiels dargestellt. Der Fokus liegt dabei insbesondere auf den ersten beiden Teilschritten der in Abschnitt 1.2.2 beschriebenen BCI-Methode, d.h. die Identifizierung und Bewertung der relevanten Informationen und deren Übertragung auf einen ungerichteten Graphen. Beide Schritte bilden den Ausgangspunkt für die sich anschließende Abgrenzung von Komponenten mit den Methoden der Komponentenfindung.

1.3.1 Einführung Fallbeispiel OnlineCars

Das Fallbeispiel soll einen überschaubaren, nachvollziehbaren Sachverhalt darstellen sowie die vorhandenen Informationen, welche sich in Form von konzeptuellen Modellen im SOM-Ansatz beschreiben lassen, darlegen. Um die Vollständigkeit bzgl. der abzubildenden Informationen sicherzustellen, orientiert sich die Modellierung des Fallbeispiels strikt an dem im SOM-Vorgehensmodell spezifizierten schrittweisen Vorgehen (vgl. Abschnitt 1.2.3). Als Anwendungsfall aus dem betrieblichen Kontext wird stark vereinfachend die Vermietung von Elektrofahrzeugen betrachtet. Aus Sicht des Objektsystems handelt es sich um einen Verleihservice (OnlineCars), bei dem Kunden über ein eigenes Internetportal Fahrzeuge mieten können. Der Verleihservice übernimmt die komplette Auftragsabwicklung von der Auftragsannahme über die Bereitstellung der Fahrzeuge bis hin zur Rechnungsstellung. Abb. 1.3 fasst diese Zusammenhänge auf der Geschäftsprozessebene strukturorientiert in Form mehrerer IAS zusammen. Ausgehend von einem initialen IAS wurden hierbei sukzessive Objekt- und Transaktionszerlegungen gemäß den Zerlegungsregeln des SOM-Ansatzes durchgeführt (Ferstl und Sinz 2013). Das Ergebnis bildet das finale IAS.

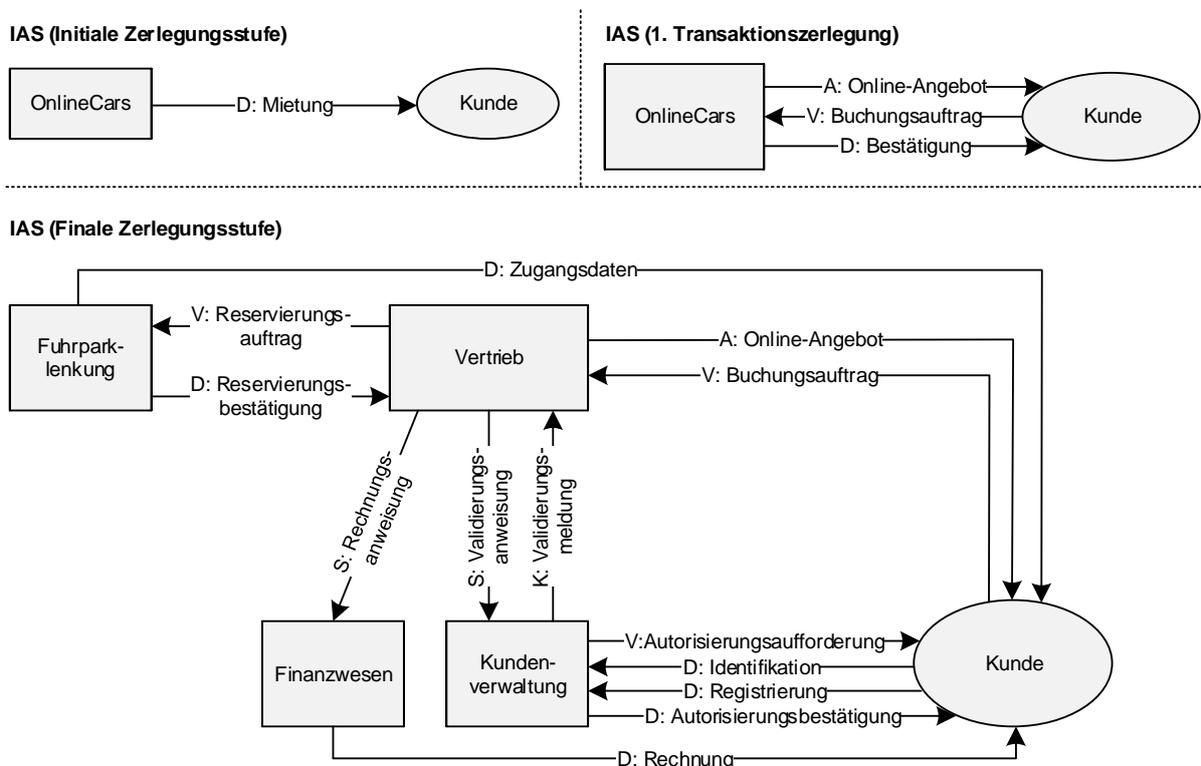


Abb. 1.3 IAS des Fallbeispiels

Während der Vertrieb dem Finanzwesen und der Kundenverwaltung in Form einer hierarchischen Koordination übergeordnet ist, erfolgt die Koordination zwischen dem Vertrieb und der Fuhrparklenkung sowie zwischen dem Vertrieb und dem Umweltobjekt Kunde auf Basis des

Verhandlungsprinzips. Der Verleihservice erstellt bei Vorliegen eines Buchungsauftrags als Leistung neben der Autorisierungsbestätigung und der Rechnung auch die Zugangsdaten und übergibt diese an den Kunden. Die Leistungen werden nach Beauftragung bzw. Anweisung des Vertriebs von unterschiedlichen Objekten erbracht. Der Kunde übermittelt seinerseits als Leistung aufgrund einer Autorisierungsaufforderung seine Registrierungs- bzw. Identifikationsdaten. In Ergänzung zum IAS beschreibt der Ausschnitt des VES in Abb. 1.4 das Verhalten der betrieblichen Objekte im Geschäftsprozess. Als Beispiel löst der Empfang eines Buchungsauftrags durch den Vertrieb über ein objektinternes Ereignis die Sendung eines Validierungsauftrags an die Kundenverwaltung aus. Dem VES kann zudem entnommen werden, dass sich der Kunde nach einer Autorisierungsaufforderung durch die Kundenverwaltung entweder registriert oder identifiziert. Es wird die Annahme getroffen, dass ein Kunde, der sich gerade registriert hat, automatisch identifiziert ist und unmittelbar die Reservierung gemäß seinem Auftrag veranlassen kann.

Die Ableitung der Anwendungssystemspezifikation in Form von KOS und VOS auf Basis der Geschäftsprozessmodelle IAS und VES schließt die Modellierung des Anwendungsfalls ab. In Abb. 1.5 ist das initiale sowie das resultierende konsolidierte KOS dargestellt.

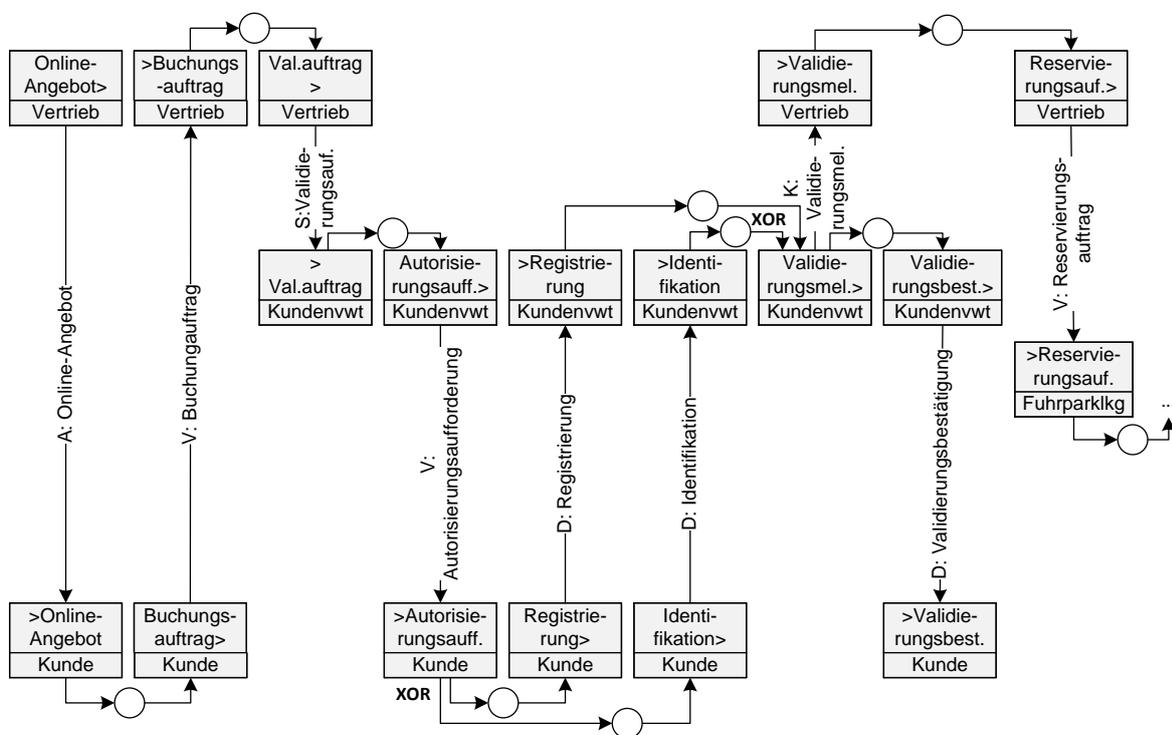


Abb. 1.4 Ausschnitt VES des Fallbeispiels (finale Zerlegungsstufe)

Entsprechend des SOM-Ansatzes werden bei der Konsolidierung diejenigen KOTs entfernt, die mit nicht-automatisierten Aufgaben und Transaktionen korrespondieren (Ferstl und Sinz 2013). Zudem werden KOTs, die sich in ihren Attributen und/oder Operatoren weitestgehend decken aus Redundanzgründen zusammengefasst. Die bei der Konsolidierung durchgeführten Änderungen sind in Tab. 1.2 festgehalten. Neben der Zusammenfassung und Umbenennung 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*) unterschieden sowie die Kardinalitäten der Beziehungen ermittelt. Im Anwendungsfall stellt die *Autorisierung* eine Generalisierung der KOTs *Registrierung* und *Identifikation* dar, die einzelnen Positionen einer Rechnung (*RechPos*) sind über die Aggregationsbeziehung einer bestimmten *Rechnung* zugeordnet.

Das für jedes betriebliche Objekt erstellte VOS wird in einer integrierten Darstellung in Abb. 1.6 zusammengeführt. Analog zum Vorgehen bei der Erstellung des 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.

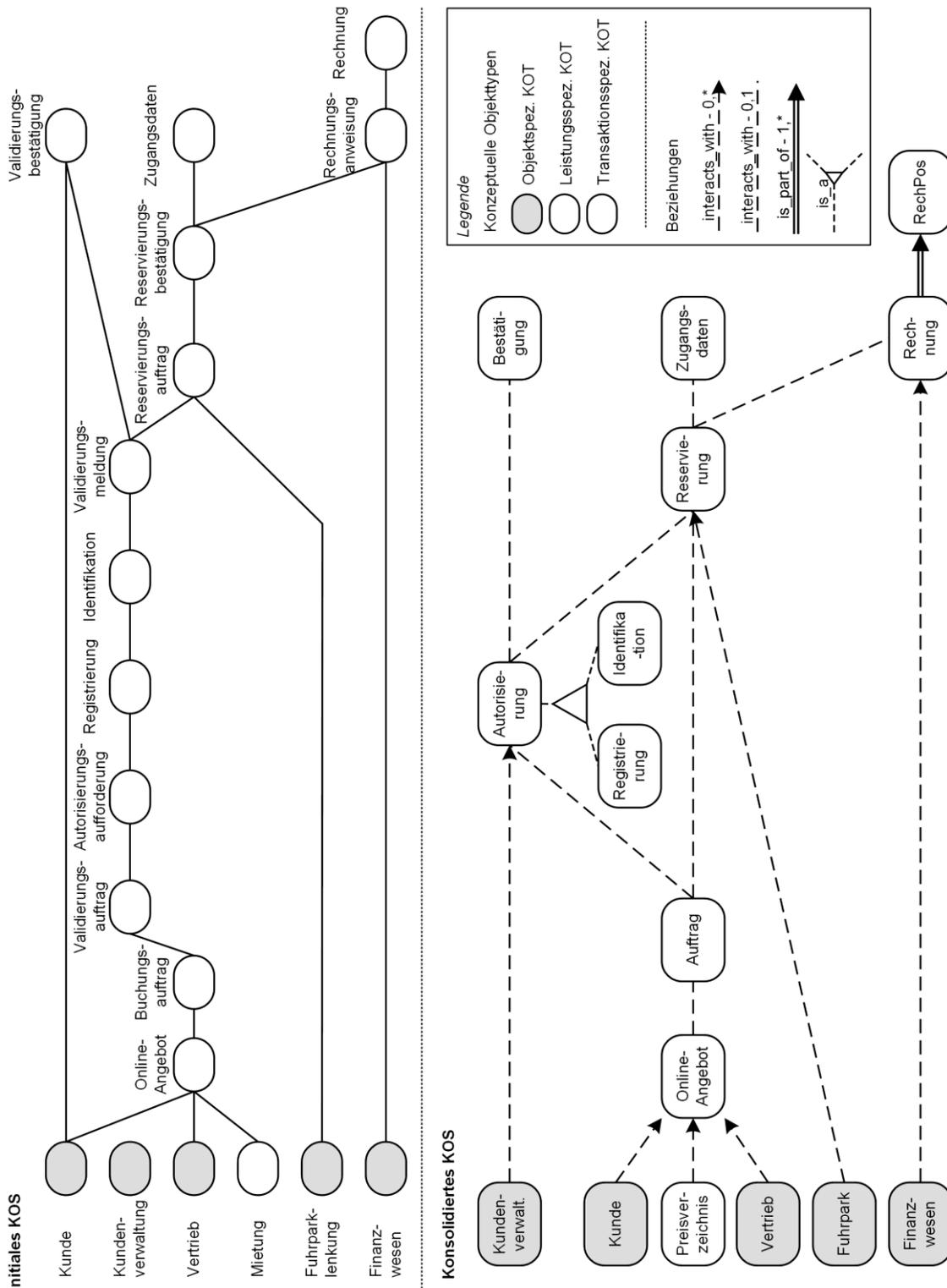


Abb. 1.5 Initiales und konsolidiertes KOS des Fallbeispiels

Tab. 1.2 Vorgenommene Änderungen bei der Konsolidierung des KOS

	<i>Initiales KOS</i>	<i>Konsolidiertes KOS</i>
<i>Objektspezifische KOTs</i>	Vertrieb	Vertrieb
	Kunde	Kunde
	Kundenverwaltung	Kundenverwaltung
	Fuhrparklenkung	Fuhrpark
	Finanzwesen	Finanzwesen
<i>Leistungsspezifische KOTs</i>	Mietung	Preisverzeichnis
<i>Transaktionsspezifische KOTs</i>	Online-Angebot	Online-Angebot
	Buchungsauftrag	Auftrag
	Validierungsbestätigung	Bestätigung
	Validierungsauftrag Autorisierungsaufforderung Validierungsmeldung	Autorisierung
	Registrierung	Registrierung
	Identifikation	Identifikation
	Reservierungsauftrag Reservierungsbestätigung	Reservierung
	Zugangsdaten	Zugangsdaten
	Rechnungsanweisung Rechnung	Rechnung

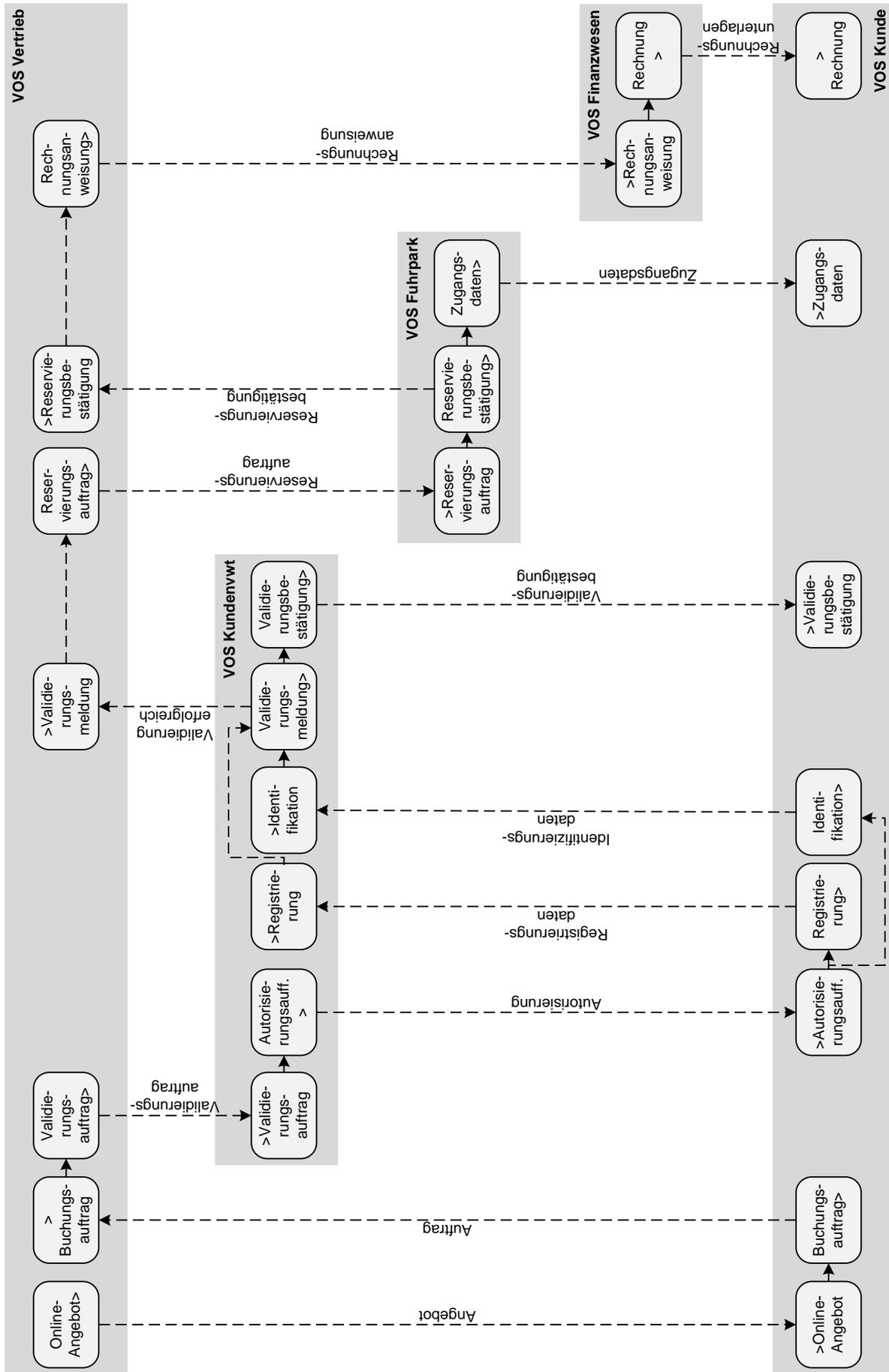


Abb. 1.6 Konsolidiertes VOS des Fallbeispiels

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.

1.3.2 Identifizierung und Bewertung der relevanten Informationen

Mit Blick auf die Ergebnisse der Modellierung des Anwendungsfalls wird bereits deutlich, dass sich die Frage, welche konkreten Informationen bei der Komponentenfindung mittels der BCI-Methode berücksichtigt werden sollen, nicht direkt beantworten lässt. 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.

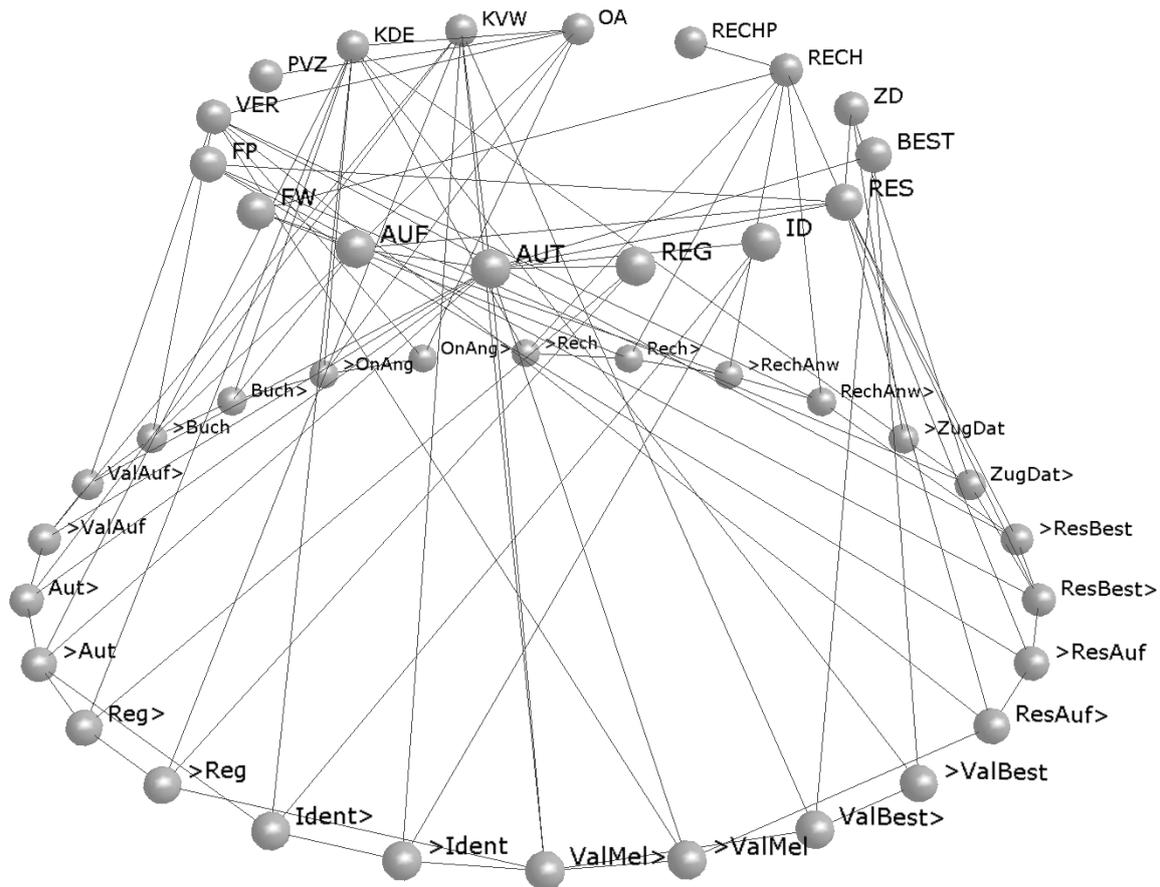
Es ist somit zu klären, welche Modelle des SOM-Ansatzes sich am besten für die Komponentenfindung eignen. Objekte sowie deren Beziehungen sind in den strukturorientierten Modellen IAS und KOS abgebildet, Vorgänge bzw. Aufgaben in den verhaltensorientierten Modellen VES und VOS. Wie bereits in Abschnitt 1.2.3 angemerkt, werden im IAS und VES auch nicht-automatisierbare bzw. nicht automatisierte Elemente abgebildet. Eine Komponentenfindung auf Basis dieser Modelle kann daher Komponenten hervorbringen, welche von einem Anwendungssystem nur teilweise bzw. überhaupt nicht realisiert werden können. Bei KOS und VOS entfällt diese Problematik, da hier eine Konsolidierung in Verbindung mit der Entfernung nicht-automatisierbarer Elemente vorgenommen wird. Darüber hinaus ergibt sich bei der Nutzung von KOS und VOS kein Informationsverlust gegenüber dem IAS bzw. VES, da KOS und VOS aus IAS und VES abgeleitet werden und diese hinsichtlich der Anwendungssystemspezifikation verfeinern und ergänzen (Ferstl und Sinz 2013). In Bezug auf das Fallbeispiel werden KOS und VOS daher als Ausgangsbasis für die Komponentenfindung definiert.

Der Entscheidung für die Nutzung von KOS und VOS schließt sich die Bewertung der darin enthaltenen Beziehungen im Hinblick auf ihre Relevanz an. Für die Komponentenfindung werden die Aggregations- und Generalisierungsbeziehungen im KOS aufgrund der engen Kopplung von KOTs relevanter eingestuft als die einfachen Interaktionsbeziehungen zwischen KOTs. Mit Blick auf das KOS ergibt sich theoretisch eine weitere Differenzierungsmöglichkeit, da bei den KOTs zwischen objektspezifischen, leistungsspezifischen und transaktionsspezifischen KOTs differenziert wird. Eine Interaktionsbeziehung zwischen einem leistungsspezifischen und einem transaktionsspezifischen KOT könnte für die Komponentenfindung daher bspw. relevanter eingestuft werden, als eine Interaktionsbeziehung zwischen zwei transaktionsspezifischen KOTs. Jedoch lässt sich nur schwer eine logische Begründung für die Einstufung dieser Relevanz geben. Im Rahmen des Fallbeispiels werden Interaktionsbeziehungen zwischen den KOTs daher nicht weiter differenziert.

Im VOS wiederum werden grundsätzlich nur Interaktionsbeziehungen dargestellt, jedoch kann in Anlehnung an das VES zwischen objektinternen (objektinterne Ereignisse) und objektexternen Beziehungen (Transaktionen) differenziert werden. 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. In Bezug auf die objektexternen Beziehungen des VOS ergibt sich theoretisch ebenfalls eine weitere Differenzierungsmöglichkeit, da jede objektexterne Beziehung als Transaktion einer bestimmten Koordinationsform (z.B. Anbahnung, Vereinbarung, Durchführung) entspricht. Jedoch lässt sich auch hier nur schwer eine logische Aussage bzgl. der Relevanz der unterschiedlichen Koordinationsformen für die Komponentenfindung treffen. Für das Fallbeispiel werden objektexterne Beziehungen daher ohne weitere Unterscheidung unterhalb den objektinternen Beziehungen eingeordnet.

1.3.3 Übertragung der identifizierten Informationen auf einen Graphen

Vor der Anwendung verschiedener Start- und Verbesserungsheuristiken müssen die als relevant identifizierten Informationen aus KOS und VOS auf einen Graphen übertragen werden. KOTs und VOTs repräsentieren Elemente und werden entsprechend der BCI-Methode als Knoten im Graphen abgebildet. Die Beziehungen zwischen den KOTs und zwischen den VOTs werden als ungerichtete Kanten dargestellt. In Bezug auf den SOM-Ansatz ergibt sich bei der Übertragung der Informationen aus KOS und VOS die Besonderheit, dass hierbei zwei nicht miteinander verbundene Teilgraphen entstehen, da die Zuordnung zwischen KOTs und VOTs nicht explizit in einem der Modelle angegeben wird. Um einen für die Komponentenfindung notwendigen, zusammenhängenden Graphen zu erhalten, muss daher geklärt werden, welche KOTs mit welchen VOTs in Beziehung stehen.



Abkürzungen	
KOT	VOT
AUF = Auftrag	Aut = Autorisierungsaufforderung
AUT = Autorisierung	Buch = Buchungsauftrag
BEST = Bestätigung	Ident = Identifikation
FP = Fuhrpark	OA = OnlineAngebot
FW = Finanzwesen	PVZ = Preisverzeichnis
ID = Identifikation	RECH = Rechnung
KDE = Kunde	RECHP = RechPos
KVV = Kundenverwaltung	REG = Registrierung
	RES = Reservierung
	VER = Vertrieb
	ZD = Zugangsdaten
	ResAuf = Reservierungsauftrag
	ResBest = Reservierungsbestätigung
	ValAuf = Validierungsauftrag
	ValBest = Validierungsbestätigung
	ValMel = Validierungsmeldung
	ZugDat = Zugangsdaten

Abb. 1.7 Graph der BCI-Methode

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. In einem ersten Schritt werden daher zunächst die einzelnen VOTs mit den objektspezifischen KOTs verbunden. In Bezug auf das 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*. Demzufolge kann eine Beziehung zwischen dem VOT *Online-Angebot*> und dem KOT *Vertrieb* hergestellt werden. Aufgrund der im KOS durchgeführten Konsolidierung besteht jedoch die Möglichkeit, dass KOTs umbenannt oder mit anderen KOTs zusammengefasst wurden. Um die Zuordnung dennoch zu ermöglichen, ist es notwendig, dass die durchgeführten Konsolidierungsschritte zur Nachvollziehbarkeit der Änderungen dokumentiert sind. Im vorliegenden Fallbeispiel basiert die Dokumentation der Konsolidierungsschritte auf Tab. 1.2 in Abschnitt 1.3.1. Wie sich der Tabelle entnehmen lässt, wurden in Bezug auf das Fallbeispiel keine objektspezifischen KOTs bei der Konsolidierung verändert. In einem zweiten Schritt werden die VOTs den transaktionsspezifischen KOTs zugeordnet. Auf Grundlage der Dokumentation der Konsolidierungsschritte kann 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 ist die Erstellung des Graphen abgeschlossen. Abb. 1.7 veranschaulicht das Ergebnis der Übertragung der Objekte und Beziehungen auf den Graphen.

1.4 Ergebnisse des Fallbeispiels

Die optimale Partitionierung des Graphen ist in Abb. 1.8 als Ergebnis der Komponentenfindung 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 1.3.2 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 Abb. 1.8 entnommen werden.

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 im Vergleich 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 *Kundenverwaltung*. 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 $\overline{C_{ou}(P)} \approx 0.20$ sowie entsprechend ein Kohäsionsverhältnis von $\overline{C_{oh}(P)} \approx 0.80$. 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.

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 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 und damit auch im Hinblick auf die in Abschnitt 1 dargestellten Probleme bei der Systementwicklung einen Beitrag zur Lösung zu leisten. Zudem kann in zukünftigen praxisnahen Forschungsarbeiten die Unterscheidung der Beziehungen sowie die Gewichtungen der Beziehungsarten diskutiert werden, um hierdurch eine konsolidierte Unterscheidung der Beziehungen zu erreichen. Auf Basis eines solchen konsolidierten Ergebnisses können die Systemarchitekten weitaus leichter Anpassungen entsprechend ihrer persönlichen Präferenzen vornehmen oder auch die Sensitivität der Ergebnisse prüfen. Mithilfe des im Rahmen des Beitrags verwendeten Tools, das die BCI-Methode umsetzt, ist es weiterhin möglich, den Ansatz auch in größeren Projekten zu verwenden. Insbesondere bei großen und industrienahen 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. Nicht zuletzt stellen die Ergebnisse der Durchführung der BCI-Methode auf Basis von SOM im Vergleich zu anderen konzeptionellen Modellen (EPK/ERM oder UML) eine Diskussionsgrundlage dar. Weiterhin können als Resultat solcher Diskussionen SOM-Modelle auch in anderen Komponentenfindungsverfahren verwendet werden, bspw. im Rahmen der Verwendung genetischer Algorithmen, 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 (Sommerville 2007; Zwintzsch 2005). 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.

1.6 Literatur

- Birkmeier D, Overhage S (2012) A Method to Support a Reflective Derivation of Business Components from Conceptual Models. *Information Systems and e-Business Management* 11 (3):403-435. doi:10.1007/s10257-012-0196-6
- Brown AW (2000) *Large-Scale, Component-Based Development*. Object and Component Technology Series. Prentice Hall, Upper Saddle River, NJ
- Cheesman J, Daniels J (2001) *UML Components: A Simple Process for Specifying Component-Based Software*. The Component Software Series. Addison-Wesley, Upper Saddle River, NJ
- Ferstl OK, Sinz EJ (1990) Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell. *WIRTSCHAFTSINFORMATIK* 32 (6):566-581
- Ferstl OK, Sinz EJ (1991) Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM). *WIRTSCHAFTSINFORMATIK* 33 (6):477-491
- Ferstl OK, Sinz EJ (1995) Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. *WIRTSCHAFTSINFORMATIK* 37 (3):209-220
- Ferstl OK, Sinz EJ (2013) *Grundlagen der Wirtschaftsinformatik*. 7. Aufl. Oldenbourg, München
- Garey M, Johnson D, Stockmeyer L (1974) Some Simplified NP-Complete Problems. In: *Proceedings of the 6th Annual ACM Symposium on Theory of Computing*, S 47–63
- Gorton I (2011) *Essential Software Architecture*. 2. Aufl. Springer, Heidelberg
- Herzum P, Sims O (2000) *Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise*. OMG Press Books. John Wiley & Sons, New York, NY
- Kernighan B, Lin S (1970) An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical Journal* 49:291-307
- Lim WC (1998) *Managing Software Reuse - A Comprehensive Guide to Strategically Reengineering the Organization for Reusable Components*. Prentice Hall, Upper Saddle River, NJ
- McDonald M, Begin J, Fortino S (2009) *Meeting the challenge: the 2009 CIO agenda*. Gartner, Inc.
- Parnas DL (1972) On the Criteria to be Used in Decomposing Systems into Modules. *Communications of the ACM* 15 (12):1053-1058
- Rautenstrauch C, Turowski K (2001) Common Business Component Model (COBCOM): Generelles Modell komponentenbasierter Anwendungssysteme. In: 5. Internationale Tagung Wirtschaftsinformatik, S 681-695.
- Sharp J, Ryan S (2005) A Review of Component-Based Software Development. In: *Proceedings of the 26th International Conference on Information Systems (ICIS)*
- Sommerville I (2007) *Software Engineering*. 8. Aufl. Pearson Education, Harlow
- Szyperski C, Gruntz D, Murer S (2002) *Component Software: Beyond Object-Oriented Programming*. 2. Aufl. Addison-Wesley, Harlow
- Turowski K (2003) *Fachkomponenten: Komponentenbasierte betriebliche Anwendungssysteme*. Magdeburger Schriften zur Wirtschaftsinformatik. Shaker, Aachen
- Vitharana P, Jain H, Zahedi F (2004) Strategy-Based Design of Reusable Business Components. *IEEE Trans Syst Man Cybern* 34 (4):460-474
- Vitharana P, Zahedi F, Jain H (2003) Knowledge-Based Repository Scheme for Storing and Retrieving Business Components: A Theoretical Design and an Empirical Analysis. *IEEE Transactions on Software Engineering* 29 (7):649-664

Wallnau KC, Hissam SA, Seacord RC (2002) Building Systems from Commercial Components. Series in Software Engineering. Addison-Wesley, Upper Saddle River, NJ
Zwintzsch O (2005) Software-Komponenten im Überblick: Einführung, Klassifizierung & Vergleich von JavaBeans, EJB, COM+, .Net, CORBA, UML 2. W3L, Herdecke